



Distributed source coding: New tools and application to video compression

Denis Kubasov

► To cite this version:

Denis Kubasov. Distributed source coding: New tools and application to video compression. Signal and Image Processing. Université de Rennes 1, 2008. English. NNT: . tel-01132197

HAL Id: tel-01132197

<https://hal.science/tel-01132197>

Submitted on 16 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre: 3872

THÈSE

présentée

devant l'Université de Rennes 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE RENNES 1
Mention INFORMATIQUE

par

Denis KUBASOV

Équipe d'accueil : Temics - IRISA

École Doctorale : Matisse

Composante universitaire : IFSIC

Titre de la thèse :

*Codage de sources distribuées :
nouveaux outils et application à la compression vidéo*

Soutenue le 4 décembre 2008 devant la commission d'examen

M. :	Joseph	RON SIN	Président
MM. :	Béatrice	PESQUET-POPESCU	Rapporteurs
	Michel	KIEFFER	
MM. :	Aline	ROUMY	Examineurs
	Luis	TORRES	
	Christine	GUILLEMOT	Directeur de thèse

In theory, there is no difference between theory and practice.

But, in practice, there is.

– Jan L.A. van de Snepscheut

Remerciements

Ce travail a été réalisé au sein de l'équipe-projet TEMICS à l'INRIA Rennes, et je remercie en premier lieu Christine GUILLEMOT, Directrice de recherche à l'INRIA, et responsable du projet TEMICS, pour m'avoir accueilli dans son équipe, et aussi pour m'avoir guidé et motivé dans mes recherches pendant ces trois ans. Je tiens aussi lui exprimer ma reconnaissance pour sa patience pendant nos discussions fructueuses, bien que parfois un peu longues.

Je tiens à remercier les membres de ce jury pour avoir accepté de juger ce travail. Notamment, je remercie chaleureusement Joseph RONSIN, Professeur à l'INSA à Rennes, qui m'a fait l'honneur de présider ce jury de thèse, mais également Béatrice PESQUET-POPESCU, Professeur à TELECOM ParisTech, et Michel KIEFFER, Maître de conférence à l'Université de Paris XI, pour le travail considérable qu'ils ont effectué en tant que rapporteurs.

Je remercie Luis TORRES, Professeur à l'Universitat Politècnica de Catalunya à Barcelone, d'avoir accepté de participer dans ce jury, mais aussi pour l'ambiance chaleureuse et fructueuse pendant les réunions du projet DISCOVER, auquel j'ai eu la chance de participer.

Je remercie Aline ROUMY, Chargée de recherche à l'INRIA, d'avoir accepté de juger ce travail, de m'avoir encadré pendant la dernière année, pour ses idées lumineuses, mais surtout pour ses conseils précieux concernant le manuscrit et l'exposé.

Je remercie mes collègues et mes amis, les membres de l'équipe TEMICS, qui, d'une manière ou d'une autre, m'ont conduit vers l'aboutissement de ce travail. En particulier, un grand merci à Angélique pour sa bonne humeur et les "corrections de la grand-mère", et également à Simon, à qui je dois mes connaissances "idoines" en français.

Je remercie également mes collègues du projet DISCOVER, notamment Yuri, Xavi, Sven, mais aussi toutes et tous les autres, pour une collaboration efficace, et pour les moments agréables pendant les réunions DISCOVER.

Je tiens à remercier Yuri BAYAKOVSKI, Professeur à l'Université d'État de Moscou, qui a beaucoup contribué à ce que cette thèse soit possible pour moi, et aussi Dmitriy VATOLIN et Alexey LUKIN, Chercheurs à l'Université d'État de Moscou, qui m'ont initié dans ce domaine de recherche.

Je remercie Daria, pour avoir toujours été à mes côtés pendant tout ce temps-là, pour m'avoir soutenu, aidé et motivé, pour son courage et pour son esprit voyageur, qui ont remplis ces années de sensations variées et agréables.

Enfin, je remercie ma famille pour le soutien, et surtout pour la foi inconditionnelle en moi, qui m'a permis de trouver des forces pour ce travail.

Table des matières

Table des matières	1
I Problématique et résumé des contributions	7
Introduction	9
1 Cadre théorique	13
1.1 Quantité d'information et l'entropie de données	14
1.2 Codage de source	16
1.2.1 Compression sans perte	17
1.2.1.1 Codes à longueur variable, code de Huffman	17
1.2.1.2 Codage arithmétique	19
1.2.2 Compression avec pertes	19
1.2.2.1 Quantification	20
1.2.2.2 Théorie débit-distorsion	21
1.2.2.3 Optimisation débit-distorsion	23
1.3 Codage de canal	25
1.3.1 Capacité de canal	26
1.3.2 Les codes turbo	27
1.3.3 Les codes LDPC	28
1.4 Codage de sources distribuées	29
1.4.1 Théorème de Slepian-Wolf	29
1.4.2 Théorème de Wyner-Ziv	31
1.5 Conclusion	33
2 Codage de sources et de vidéo distribués : état de l'art et résumé des contributions	35
2.1 Codage de Slepian-Wolf des sources discrètes	36
2.1.1 Techniques de codage de Slepian-Wolf utilisant les codes de source	36
2.1.1.1 Technique de codage de Al Jabri et Al-Issa	36
2.1.1.2 Technique de codage de Zhao et Effros	38

2.1.2	Techniques de codage de Slepian-Wolf utilisant les codes canal . . .	41
2.1.2.1	Codage distribué avec des syndromes - DISCUS	42
2.1.2.2	Technique de codage utilisant deux codes turbo poin- çonnés	43
2.1.2.3	Technique de codage avec des machines à nombre d'états fini	44
2.1.2.4	Technique de codage avec l'information adjacente en utilisant des codes turbo	45
2.1.2.5	Technique de codage avec l'information adjacente en utilisant des codes LDPC	46
2.2	Codage de Wyner-Ziv des sources Gaussiennes	49
2.2.1	Techniques de codage avec le quantificateur en lattice emboîtée . .	49
2.2.1.1	Exemple simple	50
2.2.1.2	Technique de codage de Zamir <i>et al.</i>	51
2.2.2	Techniques de codage DISCUS	53
2.2.3	Techniques de codage avec un quantificateur scalaire suivi par le codage de Slepian-Wolf	55
2.2.3.1	Schéma de codage de Wyner-Ziv par Aaron et Girod . .	55
2.2.3.2	Quantification optimale	56
2.3	Codage distribué de vidéo	58
2.3.1	Système de codage vidéo distribué PRISM	59
2.3.1.1	Processus de codage	59
2.3.1.2	Processus de décodage	62
2.3.2	Système de codage vidéo distribué de Stanford	62
2.3.2.1	Processus de codage	63
2.3.2.2	Processus de décodage	64
2.3.3	Diverses améliorations du système de codage de Stanford	66
2.3.3.1	Choix adaptatif de la taille de GOP	66
2.3.3.2	Quantification	66
2.3.3.3	Codage de Slepian-Wolf	67
2.3.3.4	Amélioration de l'information adjacente	67
2.3.3.5	Modélisation de corrélation	68
2.3.3.6	Reconstruction	69
2.3.3.7	Le choix du mode de codage par blocs	69
2.3.3.8	Utilisation de la voie de retour	69
2.3.3.9	Estimation de taux d'erreurs	70
2.3.3.10	Conclusion	70
2.3.4	Système de codage vidéo de Wyner-Ziv DISCOVER	71
2.3.4.1	Classification d'images	71
2.3.4.2	Transformée et quantification	73
2.3.4.3	Le codage par plan de bits	74

2.3.4.4	Estimation de débit minimal	75
2.3.4.5	Génération de l'information adjacente	77
2.3.4.6	Modélisation de la corrélation	79
2.3.4.7	Reconstruction	80
2.3.4.8	Résultats des simulations	81
2.4	Résumé des contributions	81
2.4.1	Extraction de l'information adjacente	82
2.4.2	Problèmes liés à la modélisation de la corrélation	83
2.4.3	Codage de Wyner-Ziv des sources avec mémoire	85
2.5	Conclusion	86

II Contributions 87

3	Side information extraction	89
3.1	Mesh-based motion estimation and compensation	92
3.1.1	Mesh construction in the base frame	93
3.1.2	Mesh motion parameters estimation	93
3.1.3	Bidirectional motion compensated frame interpolation	95
3.1.4	Experimental results	96
3.1.5	Conclusion on mesh-based motion estimation and compensation	97
3.2	Hybrid mesh-based/block-based interpolation	97
3.2.1	Hybrid interpolation in pixel domain	101
3.2.2	Hybrid interpolation in transform domain	103
3.2.3	Improving hybrid interpolation with information of the decoder	105
3.2.3.1	Combination rule using previously decoded bands . . .	105
3.2.3.2	Combination rule using previously decoded bands and bit planes	108
3.2.3.3	Discussion on updating the hybrid interpolation	109
3.2.4	Hybrid SI extraction using oracle	110
3.2.5	Experimental results	110
3.2.6	Conclusion on hybrid SI extraction	112
3.3	Multi-hypotheses decoding	112
3.3.1	Multi-hypotheses virtual correlation channel initialisation	113
3.3.2	Multi-hypotheses reconstruction	114
3.3.3	Experimental results	115
3.3.4	Conclusion on multi-hypotheses decoding	117
3.4	Spatial side information	117
3.4.1	Spatial SI construction	118
3.4.2	Performance analysis	119
3.4.3	Correlation model verification for spatial SI	120
3.4.4	A new quality measure for side information	123

3.4.5	Conclusion on spatial side information	125
3.5	Side information denoising	126
3.5.1	Denoising by wavelet thresholding	126
3.5.2	Denoising using adaptive support transforms	130
3.6	Conclusion	134
4	Correlation modeling in Wyner-Ziv video coding	137
4.1	DISCOVER correlation model	138
4.2	Updating the correlation model during decoding	141
4.3	Minimal MSE reconstruction	146
4.3.1	Closed-form expressions for the single side information case . . .	148
4.3.2	Closed-form expressions for multihypotheses decoding	150
4.3.3	Experimental results	154
4.4	Hybrid encoder/decoder rate control solution	154
4.4.1	Estimation of the minimal theoretical rate	156
4.4.2	Robust bit error rate estimation	159
4.4.3	Experimental results	161
4.5	Rate-distortion optimisation	161
4.5.1	Quantiser design	164
4.5.2	Distortion-rate function estimation	165
4.5.3	Optimisation	166
4.5.4	Experimental results	167
4.6	Robust estimation of correlation model parameter at the encoder	167
4.7	Mixed intra/WZ encoding mode	172
4.7.1	An overview of the proposed approach	173
4.7.2	The pattern of frame partitioning	174
4.7.3	Threshold selection	175
4.7.4	WZ frames mixed encoding	177
4.7.5	WZ frames mixed decoding	177
4.7.6	Experimental results	178
4.8	Conclusion	178
5	Source coding aspects of Wyner-Ziv video coding	183
5.1	Exploiting memory on the pixel level	183
5.1.1	Lifting wavelet decomposition	184
5.1.2	Oriented wavelet transform	185
5.1.3	Experimental results with the DISCOVER codec	187
5.2	Distributed DPCM	189
5.2.1	DPCM	190
5.2.2	Prediction in Wyner-Ziv coding	191
5.2.3	Application to video	193
5.2.4	Experimental results	194

5.2.4.1	Experimental results with a Gauss-Markov process . . .	194
5.2.4.2	Experimental results with real video data	196
5.3	Huffman coding of quantisation indices	196
5.3.1	Integrating VLC support in DISCOVER coding scheme	197
5.3.2	Experimental results	198
5.4	Conclusion	200
Conclusion and perspectives		201
Publications		205
Glossary		207
Bibliography		218
Table of figures		219
Table des matières		

Première partie

**Problématique et résumé des
contributions**

Introduction

L'essor de l'informatique durant les quelques dernières décennies fit entrer l'humanité dans une nouvelle ère numérique. Aujourd'hui, la plupart des informations, ou des données, que l'on est amené à manipuler dans notre vie quotidienne, est stockée, traitée et représentée sous forme numérique, c'est-à-dire sous une forme compréhensible par l'ordinateur.

L'information audiovisuelle constitue une grande partie de ces données, et pour réduire le coût de stockage, cette information doit être comprimée. On utilise des algorithmes de compression et de décompression très souvent sans s'en rendre compte : en écoutant de la musique sur un CD, en regardant la télévision, en imprimant ses photos de vacances, etc. Dans chaque cas, une technique de compression avancée nous permet soit de stocker plus d'information sur un même support (CD, disque dur), soit d'avoir une meilleure qualité du son ou de l'image stocké sur un même support.

Les techniques de compression des données audiovisuelles se sont beaucoup améliorées pendant la dernière décennie permettant la banalisation des appareils multimédias grand public (baladeurs MP3, baladeurs DVD, appareils photo numériques). Il y a peu de gens aujourd'hui qui n'ont jamais entendu les noms de standards de compression comme MP3, MPEG ou JPEG.

Intéressons-nous au paradigme actuel de compression des données vidéo. En règle général, la technique de codage se base sur les outils suivants :

- un estimateur de mouvement pour exploiter la redondance temporelle (entre les images) ;
- une transformée (typiquement la transformée en cosinus discrète, DCT) pour exploiter la redondance spatiale (dans l'image) ;
- un quantificateur des coefficients de transformée pour exploiter l'imperfection d'un oeil humain ;
- un codeur entropique pour exploiter la redondance statistique dans les coefficients de quantification.

L'estimateur de mouvement joue ici le rôle d'un prédicteur de l'image à partir des autres images, c'est pourquoi ce paradigme est souvent appelé "prédictif". Notons que les codeurs vidéo actuels comprennent également un mode de prédiction spatiale, c'est-à-dire de prédiction d'un bloc à partir des autres blocs dans la même image, le terme "prédictif" n'est donc pas spécialement lié à l'estimation de mouvement. Cette

prédiction est faite naturellement au codeur car il dispose de l'image à prédire ainsi que les images servant de référence. Bien qu'en général la prédiction est l'étape qui permette de gagner le plus en terme de compression, c'est aussi l'étape la plus complexe. Par conséquent, dans le paradigme actuel le codeur est beaucoup plus complexe que le décodeur.

Ce déséquilibre convient parfaitement au cas de la liaison descendante. Dans ce type de scénario l'information est codée par un émetteur aux grandes ressources et peut être ensuite décodée plusieurs fois (par exemple la diffusion vidéo, stockage de vidéo etc). Cependant, aujourd'hui ce modèle est remis en question par l'utilisation de plus en plus répandue des réseaux sans fil, ainsi que des appareils à consommation faible d'énergie avec des capteurs vidéo intégrés, soit en liaison montante. Ce dernier suppose un codeur à complexité limitée par les capacités d'un appareil (énergie, mémoire), tandis que le décodeur (souvent installé sur une machine puissante de type serveur) n'a pas de contraintes de complexité (dans la limite du raisonnable).

C'est pourquoi les regards se sont tournés ces dernières années vers un nouveau paradigme de codage appelé le codage distribué. Ce paradigme se base sur les résultats théoriques établis par Slepian et Wolf [SW73] en 1973 pour le codage sans perte, puis étendus par Wyner et Ziv [WZ76] au cas du codage avec pertes en 1976. Ces résultats montrent qu'en théorie (sous certaines conditions) l'exploitation de la redondance peut être déplacée du codeur au décodeur sans que cela impacte la performance totale du système de codage. Ainsi pour le codage vidéo, l'étape de prédiction avec estimation de mouvement n'est plus indispensable au codeur : la redondance temporelle peut être exploitée par le décodeur uniquement.

Les avantages principaux d'un système de codage vidéo distribué par rapport aux systèmes conventionnels (prédictifs) peuvent être résumés comme suit :

- répartition flexible de la complexité entre le codeur et le décodeur ;
- résistance naturelle aux erreurs de transmission ;
- scalabilité du système (car un niveau ne dépend pas du décodage précis des niveaux précédents) ;
- application naturelle à la compression multivue, où des images, des vidéos de la même scène sont filmées de points de vue différents.

Les domaines potentiels d'utilisation de ce paradigme distribué incluent la visio-phonie, la surveillance vidéo avec des caméras sans fil, la stéréovision (pour le multivue) etc, à conditions que les algorithmes réels de codage puissent réaliser des performances comparables à ceux des algorithmes prédictifs. Cependant, actuellement il y a une divergence entre la théorie (qui prétend l'égalité de deux paradigmes) et la pratique. En effet, les performances en terme de débit-distorsion des systèmes réels du codage vidéo distribué sont très éloignées de ceux des systèmes prédictifs. Réduire cet écart est un enjeu important dont l'avenir de ce nouveau paradigme dépend beaucoup.

C'est dans cet esprit que le projet Européen DISCOVER [dis] a démarré en sep-

tembre 2005, et auquel nous avons eu la chance de participer et de contribuer. L'objectif principal de ce projet était de chercher des solutions et développer des outils performants, mieux adaptés au paradigme distribué, afin d'améliorer les performances (en terme de débit-distorsion, résistance aux erreurs etc.) de façon significative. Ce but a été atteint, et le projet a été conclu en décembre 2007 avec succès.

La plupart des travaux présentés dans cette thèse ont été effectués dans le cadre de ce projet. Nous nous sommes concentrés essentiellement à l'amélioration des performances du système de codage en terme de débit-distorsion dans le contexte monovue, sans aborder ni le cas multivue, ni la résistance aux erreurs. Trois axes principaux de recherche ont été identifiés, qui chacun, correspond à un chapitre du manuscrit : l'extraction de l'information adjacente, la modélisation de la corrélation, et les aspects codage de source du codage vidéo Wyner-Ziv.

Dans le chapitre 3 nous présentons nos études concernant l'extraction de l'information adjacente (IA). C'est l'étape de décodage qui correspond à l'étape de prédiction dans les codeurs vidéo conventionnels. Nous considérons dans un premier temps une méthode d'estimation de mouvement basée sur une représentation par maillages déformables, pour l'extraction de l'IA. Ensuite, nous développons une approche hybride qui permet de combiner plusieurs IA pour un décodage plus performant. Nous comparons cette approche avec la méthode dite le "décodage multihypothèse".

Nous considérons également une méthode de construction de l'IA spatiale. Si les améliorations ne sont pas significatives, cette étude nous permet néanmoins de déduire une mesure de qualité pour l'IA. Nous proposons ensuite d'améliorer la qualité de l'IA temporelle à l'aide de méthodes de débruitage d'images.

Le chapitre 4 est consacré aux problèmes liés au modèle de corrélation entre l'IA et l'image originale. Une estimation fine des paramètres du modèle est essentielle pour pouvoir exploiter pleinement la corrélation temporelle (comme le fait un codeur prédictif). Ainsi nous proposons deux méthodes pour améliorer cette estimation. La première est basée sur la technique dit Expectation-Maximisation (EM), et sert à raffiner l'estimation du paramètre pendant le décodage itératif des codes turbo. La deuxième est utilisée par le codeur dans les cas où ce dernier a besoin d'une estimation plus précise que celle que l'on peut obtenir au décodeur.

Le problème du contrôle de débit est également abordé dans le chapitre 4. Nous proposons une approche dite hybride, où l'estimation du débit est faite à la fois au codeur et au décodeur. Cette solution permet d'effectuer un contrôle assez fin, tout en diminuant le nombre de requêtes de bits supplémentaires et ainsi la complexité du décodeur.

Finalement, nous proposons deux méthodes pour optimiser les performances du codage. La première consiste à optimiser le nombre de niveaux de quantification des images Wyner-Ziv (c'est-à-dire les images codées avec le paradigme distribué), en fonction du niveau de corrélation temporelle. Cette optimisation est réalisée sous contrainte que la qualité de l'image décodée soit proche de celle des images clés adjacentes

(les images codées avec un codeur conventionnel). Dans la deuxième approche nous proposons de coder certains blocs d'une image Wyner-Ziv avec un codeur conventionnel (exploitant la corrélation spatiale), si le niveau estimé de corrélation temporelle ne permet pas de garantir un codage efficace en mode WZ.

Les aspects du codage de source sont abordés dans le chapitre 5. Nous présentons quelques algorithmes dont le but est d'exploiter la mémoire de la source, c'est-à-dire la redondance spatiale, dans le codage Wyner-Ziv. Le problème de mémoire est partiellement résolu dans le codeur DISCOVER, qui, en appliquant une transformée DCT, partitionne l'image entière en sous-bandes de fréquences indépendantes. Cependant, il reste toujours de la mémoire entre les coefficients d'une même bande.

Le premier algorithme que nous proposons pour résoudre ce problème, est la prédiction des coefficients. Nous étendons l'algorithme DPCM au cas du codage distribué, où les coefficients de l'information adjacente doivent aussi être prédits, comme les coefficients de l'image WZ. Ensuite, nous considérons une transformée par ondelettes orientées comme alternative à la transformée DCT. Nous montrons que cette dernière permet de séparer les fréquences d'une manière plus efficace que la transformée DCT dans certains cas.

Enfin, pour exploiter la redondance statistique dans les indices de quantification, il est proposé de les coder avec un code de Huffman (dans DISCOVER ils sont codés avec un code à longueur fixe). Cette approche augmente le nombre de plans de bits par bande, mais réduit la longueur totale cumulée sur tous les plans de bits (chaque plan de bits peut avoir une longueur différente des autres avec un code à longueur variable). Des gains intéressants en terme de performances débit-distorsion sont obtenus.

Chapitre 1

Cadre théorique

La compression des données est une science relativement jeune. Il est convenu de considérer comme le point de départ la publication par Claude Shannon en 1948 de “la théorie mathématique de communication” [Sha48]. Dans cette théorie il a introduit en particulier la notion de quantité d’information $\mathbb{I}(X = x)$ d’un événement $X = x$, et de l’entropie $H(X)$ d’une variable discrète, qui définit la limite de taux de compression de X .

Outre la théorie de compression des données (souvent appelée le codage de source), Shannon a également établi les fondements de la théorie de codage de canal, en introduisant quelques notions et limitations utiles pour des problèmes de codage liés à la transmission des signaux par les canaux. Ces deux aspects constituent la majeure partie de la théorie qui de nos jours s’appelle la théorie de l’information.

Dans son oeuvre Shannon considéra aussi le cas de plusieurs sources (e.g. X et Y) encodées et décodées conjointement. Il a introduit la notion de l’entropie conjointe $H(X, Y)$ étant la borne inférieure pour le débit conjoint de transmission. Cependant, ce n’était qu’en 1973 qu’un résultat équivalent fut obtenu par Slepian et Wolf [SW73] pour le cas où le codage de X et Y se fait séparément (le décodage est toujours conjoint). Ils démontrèrent que la communication entre les encodeurs n’est pas nécessaire pour atteindre le taux de compression maximum défini par Shannon, il suffit simplement que chaque codeur connaisse les statistiques des autres sources. Le résultat de Slepian et Wolf fut étendu par Wyner et Ziv au cas des sources à valeurs continues et compression avec pertes [WZ76]. Les deux travaux ont marqué le début de la théorie du codage de sources distribuées.

Dans ce chapitre nous rappelons au lecteur les notions de base de la théorie de l’information et en particulier du codage de sources distribuées, qui sont le sujet de cette thèse. Ainsi, les bases de la théorie de la compression de données et du codage de canal seront rappelées. Ensuite, les théorèmes de Slepian-Wolf et de Wyner-Ziv seront présentés. Ce rappel a pour but d’introduire le thème de cette thèse, et ne couvre donc que quelques aspects de la théorie de l’information. Pour avoir une vision plus complète, le lecteur est invité à consulter l’ouvrage de Joy A. Thomas et Thomas Cover

[CT91], par exemple.

1.1 Quantité d'information et l'entropie de données

Dans cette section nous présentons les notions basiques de la théorie de l'information, à savoir la notion de quantité d'information et d'entropie.

Soit X une variable aléatoire discrète qui prend des valeurs x dans un alphabet fini \mathcal{A} avec la probabilité $\Pr(X = x)$. La définition formelle par Shannon est la suivante.

Définition 1.1 La quantité $\mathbb{I}(X = x)$ définie par

$$\mathbb{I}(X = x) = -\log_2(\Pr(X = x)), \quad (1.1)$$

est appelée *innovation de l'événement $X = x$* , ou la *quantité d'information associée à cet événement*.

Le choix 2 de base de logarithme est arbitraire mais commode. Avec cette base, $\mathbb{I}(X = x)$ définit comme 1 l'innovation d'un événement le plus simple qu'on puisse imaginer, i.e. une réalisation d'une variable binaire de loi uniforme. Ainsi définie, l'unité d'information s'appelle un *bit*.

Pour caractériser la variable X elle-même, et non pas une réalisation $X = x$, la notion d'entropie de X est introduite.

Définition 1.2 L'entropie $H(X)$ d'une variable aléatoire X est une espérance de l'innovation $\mathbb{I}(X = x)$, $x \in \mathcal{A}$, à savoir

$$H(X) = \sum_{x \in \mathcal{A}} \Pr(X = x) \mathbb{I}(X = x) = - \sum_{x \in \mathcal{A}} \Pr(X = x) \log_2(\Pr(X = x)). \quad (1.2)$$

Cette valeur est non-négative (nulle si et seulement si $\exists x \in \mathcal{A} : \Pr(X = x) = 1$), et s'exprime en *bits par symbole* d'alphabet \mathcal{A} .

Le premier théorème de Shannon montre que l'entropie est le nombre moyen de *bits* nécessaire pour représenter une réalisation de la variable X . Autrement dit, elle définit la borne théorique pour le taux de compression d'une séquence de symboles d'alphabet \mathcal{A} avec la fréquence d'apparition des symboles définie par la loi $\Pr(X = x)$, $x \in \mathcal{A}$.

L'entropie conjointe de plusieurs variables va aussi nous intéresser. Soit Y une deuxième variable aléatoire qui prend des valeurs y dans un autre alphabet fini \mathcal{B} avec la probabilité $\Pr(Y = y)$. Notons avec $p(x, y)$ la loi de distribution conjointe de deux variables X et Y :

$$p(x, y) = \Pr(X = x, Y = y). \quad (1.3)$$

Alors, l'innovation d'un événement $(X = x \& Y = y)$ et l'entropie conjointe de deux variables X et Y peuvent être écrites comme suit :

$$\mathbb{I}(X = x, Y = y) = -\log_2(p(x, y)), \quad (1.4)$$

$$H(X, Y) = \mathbb{E}_{p(x, y)} [\mathbb{I}(X = x, Y = y)] = - \sum_{(x, y) \in \mathcal{A} \times \mathcal{B}} p(x, y) \log_2(p(x, y)). \quad (1.5)$$

On peut aussi définir l'entropie conditionnelle $H(X|Y)$ de X sachant Y , comme l'espérance de l'innovation d'un événement $X = x$ sachant $Y = y$:

$$H(X|Y) = \mathbb{E}_{p(x, y)} [\mathbb{I}(X = x|Y = y)] = - \sum_{(x, y) \in \mathcal{A} \times \mathcal{B}} p(x, y) \log_2(p(x|y)). \quad (1.6)$$

En utilisant la définition de la probabilité conditionnelle $p(x|y)p(y) = p(x, y)$, il est facile de montrer que :

$$H(X, Y) = H(X|Y) + H(Y) = H(Y|X) + H(X). \quad (1.7)$$

Cette définition se généralise pour n variables aléatoires X_0, X_1, \dots, X_{n-1} :

$$H(X_0, X_1, \dots, X_{n-1}) = H(X_0) + H(X_1|X_0) + \dots + H(X_{n-1}|X_0, X_1, \dots, X_{n-2}). \quad (1.8)$$

L'entropie conditionnelle $H(X|Y)$ de X sachant Y représente une mesure de l'incertitude qui *reste* dans X après que Y est connue. La *réduction* de cette incertitude apportée par Y , est aussi une quantité très importante. Elle s'appelle l'*information mutuelle*, et mesure la quantité d'information qu'une variable contient à propos de l'autre, autrement dit, la quantité d'information commune aux deux variables.

Définition 1.3 L'information mutuelle entre deux variables X et Y est définie par :

$$I(X; Y) = \sum_{(x, y) \in \mathcal{A} \times \mathcal{B}} p(x, y) \log_2 \left(\frac{p(x, y)}{\Pr(X = x) \Pr(Y = y)} \right). \quad (1.9)$$

On peut déduire des formes plus commodes équivalentes à la forme (1.9) :

$$I(X; Y) = H(X) - H(X|Y) \quad (1.10)$$

$$= H(Y) - H(Y|X) \quad (1.11)$$

$$= H(X) + H(Y) - H(X, Y), \quad (1.12)$$

d'où on déduit les propriétés de l'information mutuelle (données sans preuves) :

1. $\mathbf{I(X; Y)} \geq \mathbf{0}$, car $H(X) \geq H(X|Y)$;
2. $\mathbf{I(X; Y)} \leq \mathbf{\min(H(X), H(Y))}$, car $H(X|Y) \geq 0$ et $H(Y|X) \geq 0$;
3. $\mathbf{I(X; X)} = \mathbf{H(X)}$, car $H(X|X) = 0$ (c'est pourquoi l'entropie $H(X)$ est parfois appelée l'*information propre*) ;
4. $\mathbf{I(X; Y)} = \mathbf{0}$ si et seulement si les deux variables X et Y sont indépendantes.

Ces propriétés sont illustrées sur la figure 1.1 sous forme d'un diagramme de Venn.

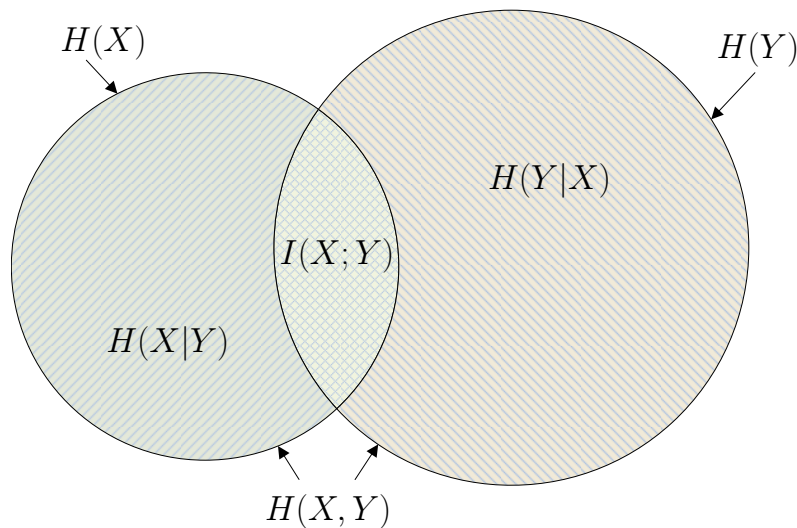


FIG. 1.1 – Relation entre l'entropie et l'information mutuelle

1.2 Codage de source

Le codage de source est une partie de la théorie d'information qui étudie les problèmes liés à la représentation la plus compacte possible des données, d'où son autre nom : la "compression de données". Dans cette section on va identifier une séquence de données avec une suite de réalisations d'une variable aléatoire, et cette variable aléatoire représente donc la *source*. On va aussi supposer que la loi de distribution de cette variable est connue à l'avance (au codeur comme au décodeur), sauf si le contraire est précisé explicitement.

De manière générale, on parle de la compression si l'espérance de la longueur du mot de code est inférieure à l'espérance de la longueur de représentation directe de symboles dans l'alphabet de mots de code (qui est souvent binaire).

Le rapport entre les longueurs de la séquence d'entrée et celle de sortie, est appelé le taux (ou facteur) de compression. En général, le but du codage de source est de construire un code qui permettrait le taux moyen de compression le plus élevé possible.

On distingue la compression sans perte (quand la reconstruction parfaite de la séquence d'entrée est garantie) et avec pertes (dans ce cas on parle de distorsion de reconstruction). Shannon démontre que le taux de compression sans perte a une limite fondamentale, et que cette limite est égale à l'entropie $H(X)$ de la source [Sha48].

Pour la compression avec pertes, il y a aussi une limite fondamentale qui dépend de la distorsion voulue. Cette limite est donc une fonction de la distorsion D . Elle s'appelle la fonction débit-distorsion $R(D)$, et définit le débit minimum théorique (e.g. en bits par symbole) nécessaire pour coder une séquence de données produite par

la source en question, pour que cette séquence soit reconstruite avec une distorsion inférieure ou égale à D .

En effet, la compression sans pertes peut se voir comme un cas particulier de la compression avec pertes avec distorsion nulle. Cela implique que $R(0) \geq H(X)$ (pour une source d'alphabet fini).

1.2.1 Compression sans perte

La compression sans pertes est le codage d'une séquence de symboles qui permet un décodage parfait. La théorie de l'information définit une limite mathématique pour le débit minimum atteignable dans ce cas-là : c'est l'entropie $H(X)$ de la source X qui a émis cette séquence.

Dans la suite, nous présentons plusieurs codes de compression sans perte. Pour plus d'information sur ce sujet, le lecteur peut lire le chapitre 5 de [CT91].

1.2.1.1 Codes à longueur variable, code de Huffman

Nous présentons tout d'abord les codes à longueur variable (CLV). L'idée générale de ce type de code est d'associer des mots de code plus courts aux symboles plus fréquents, et les mots de code plus longs aux symboles moins fréquents. En effet, l'espérance de la longueur que l'on cherche à minimiser, est donnée par :

$$L = \sum_{x_i \in \mathcal{A}} \Pr(X = x_i) l_i, \quad (1.13)$$

où l_i est la longueur du mot de code associé au symbole x_i . La loi de probabilité $\Pr(X = x_i)$ est supposée connue dans le reste de cette section.

Shannon propose un code dont l'espérance de la longueur est dans un écart de moins d'un bit de la borne théorique, i.e. :

$$H(X) \leq L_{Shan} < H(X) + 1. \quad (1.14)$$

C'est le code qui donne à chaque symbole x_i un mot de code dont la longueur dépend de sa probabilité :

$$l_i = \left\lceil \log_2 \frac{1}{\Pr(X = x_i)} \right\rceil. \quad (1.15)$$

Ce code de Shannon est facile à construire, mais il peut dépasser de 1 la longueur moyenne optimale. Quelques années plus tard, en 1952 Huffman [Huf52] a proposé un algorithme simple qui permet de construire un code optimal, i.e. avec $L^* \leq L$, où L est l'espérance de la longueur de n'importe quel autre code.

L'algorithme de construction du code de Huffman a deux étapes. La première étape, consiste à construire un arbre de symboles. Il est itératif :

1. Ranger les probabilités de symboles dans l'ordre décroissant.

TAB. 1.1 – Construction de code de Huffman

	$p_{meta}(1)$	$p_{meta}(2)$	$p_{meta}(3)$	$p_{meta}(4)$	mots de code
p_1	0.4	0.4	0.6	1.0	1
p_2	0.25	0.35	0.4		01
p_3	0.2	0.25			000
p_4	0.15				001

2. Unir les deux derniers symboles (avec les probabilités les plus faibles) en un méta-symbole avec la probabilité égale à la somme des probabilités de ces symboles.
3. S'il ne reste qu'un seul symbole, sortir de la boucle. Sinon, revenir au pas 1.

La deuxième étape, à savoir l'affectation des longueurs des mots de code, se fait dans l'ordre inverse :

1. Affecter au méta-symbole racine (qui a la probabilité 1) une longueur 0.
2. Choisir un nouveau nœud dans l'arbre avec une longueur associée ("longueur courante"). Si ce nœud est un méta-symbole composé de deux symboles ou méta-symboles (voir l'algorithme direct ci-dessus), associer au deux symboles (ou méta-symboles) constituants la longueur égale à "longueur courante + 1 bit". Sinon, propager la longueur courante jusqu'au niveau suivant d'arbre.
3. Répéter 2 jusqu'à ce que tous les symboles aient une longueur affectée.

L'algorithme est illustré dans Tab. 1.1. Les flèches illustrent la première étape (directe), tandis que les chiffres à leurs côtés représentent la deuxième étape (inverse), qui commence au méta-symbole racine avec la probabilité 1.0, et associe les mots de code en traversant l'arbre de droite à gauche.

Un code de compression sans perte est optimal si la longueur moyenne d'un mot de code est égale à l'entropie de la source. Le code de Huffman est optimal lorsque les probabilités de chaque symbole sont des puissances de 2 (une source *dyadique*). Sinon, il peut ne pas être optimal car la longueur d'un mot de code est toujours un nombre entier, ce qui produit des erreurs d'arrondi. Par exemple, aucun symbole ne peut avoir un mot de code de longueur de moins d'un bit quelle que soit sa probabilité, même si elle est très proche de 1.

Les erreurs d'arrondi peuvent être diminuées en codant tout un bloc de symboles avec un seul mot de code, cette approche étant appelée le codage de Huffman généralisé. Notons néanmoins qu'elle devient trop complexe pour des séquences de grande taille [Tja00] car le nombre de symboles dans le nouveau méta-alphabet croît exponentiellement avec la taille de la séquence.

Dans la section suivante nous présenterons une technique de codage beaucoup moins complexe qui approche efficacement la borne théorique de codage sans perte.

1.2.1.2 Codage arithmétique

L'idée originale du codage arithmétique [Ris76, Pas76] est de représenter la séquence entière d'entrée avec un seul nombre réel dans l'intervalle $[0,1)$. En effet, n'importe quelle séquence de n'importe quelle longueur peut être représentée par un tel nombre qui la distinguerait des autres séquences possibles. Par contre, si on considère vraiment toutes les séquences possibles, ce nombre peut comprendre un nombre infini de chiffres dans sa représentation binaire.

Une contrainte L sur la longueur de la séquence réduit le nombre de toutes les séquences possibles à un nombre fini. Alors, on peut sélectionner un nombre fini de sous-intervalles de $[0,1)$, dont chacun va représenter une séquence possible. De plus, on peut s'arranger pour que chaque sous-intervalle contienne un nombre réel représentable par une fraction finie dans la notation binaire. Cela va nous donner un code valable pour coder sans perte toutes les séquences de taille limitée $l \leq L$. De manière générale, en associant les séquences les plus probables avec des sous-intervalles longs, la compression est atteinte.

Cependant, cet algorithme de choix de sous-intervalles ressemble au codage de Huffman généralisé, et donc a le même problème de complexité exponentielle avec la taille de la séquence. En réalité les sous-intervalles se construisent au fur et à mesure du codage par fractionnement du sous-intervalle courant. Plus précisément, il est fractionné en sous-intervalles selon les probabilités de symboles, et le sous-intervalle qui correspond au symbole actuel d'entrée, devient le sous-intervalle courant. La complexité de cet algorithme est linéaire (avec la taille l de la séquence d'entrée).

En effet, le codage arithmétique est la technique la plus utilisée de nos jours pour la compression sans perte grâce aux avantages suivants :

- Efficacité de codage (approche la borne théorique $H(X)$ en pratique).
- Efficacité de réalisation algorithmique (il existe des réalisations très efficaces n'utilisant que des opérations arithmétiques sur les nombres entiers).
- Flexibilité (les probabilités des symboles qui définient le fractionnement de l'intervalle, peuvent varier. Cela implique un codage plus efficace de sources dont le modèle n'est pas stationnaire. Le codage arithmétique est alors dit adaptatif, voir [WNC87] par exemple).

Une réalisation très efficace d'un codeur arithmétique adaptatif est proposé dans [MSW03], et s'appelle en anglais CABAC (Context-Adaptiv Binary Arithmetic Coding). Cette réalisation est utilisée dans le standard de compression vidéo H.264 le plus performant au jour d'aujourd'hui.

1.2.2 Compression avec pertes

Les systèmes de codage avec pertes permettent une compression à taux plus élevé que $H(X)$. Pour cela, une partie de l'information et/ou la précision doivent être sacrifiées, rendant impossible la reconstruction exacte de la séquence codée (le codage

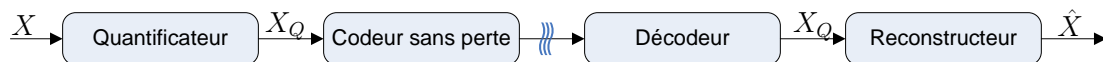


FIG. 1.2 – Schéma générale de compression avec pertes

est dit *irréversible*). Les domaines d'application de la compression avec pertes sont très différents, commençant par la numérisation des données analogiques à valeurs continues, et finissant par la compression des données discrètes. En effet, dans plusieurs applications notamment en multimédia (audio, vidéo, images, etc.), les utilisateurs peuvent tolérer une certaine dégradation de leurs données en contrepartie d'un taux de compression plus élevé. Cette dégradation est appelée la *distorsion*.

Un système de codage avec pertes peut se voir comme un système de codage sans perte, précédé par un quantificateur et suivi par un quantificateur inverse (voir la figure 1.2). La source X en passant par le quantificateur, devient la source quantifiée X_Q , qui est codée sans perte, et puis reconstruite en \hat{X} . La distorsion est définie souvent comme l'erreur moyenne quadratique entre la séquence initiale et la séquence reconstruite : $D(X, \hat{X}) = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$, bien que il y ait d'autres façons de la définir (voir la section 1.2.2.2).

C'est donc le choix de la paire quantificateur/quantificateur inverse qui définit la performance d'un système de codage avec pertes (en supposant que le codage sans perte dans le système est effectué avec un taux proche de l'entropie de la source). A la différence du codage sans perte, les performances ici sont mesurées par une paire de valeurs, à savoir le taux de compression (le terme débit est plus souvent utilisé) et la distorsion. Tout comme dans le cas du codage sans perte, une limite théorique est établie par Shannon dans ce cas-là aussi. Cette limite s'appelle la fonction débit-distorsion $R(D)$.

1.2.2.1 Quantification

Le but de la quantification est de représenter les données sur un alphabet discret. Par exemple, les valeurs réelles (dont le domaine est continu) peuvent être représentées d'une façon compacte par le numéro d'intervalle dont elles appartiennent.

On distingue trois types de quantificateurs : quantificateurs scalaires, vectoriels, et quantificateurs codés en treillis. Comme la quantification n'occupe pas une place très importante dans le champ de recherche de cette thèse, ici on va se limiter à décrire le quantificateur scalaire uniforme et le quantificateur scalaire à zone morte qui seront utilisés dans la suite. Une revue assez complète sur les techniques de quantification peut être trouvée dans [GN98].

Soit X une source réelle, dont les valeurs appartiennent à l'intervalle $[a, b]$, avec la fonction de densité de probabilité $p(x)$. Le quantificateur scalaire quantifie chaque élément x_i de X d'une façon indépendante des autres éléments x_j , $j \neq i$. Un quantificateur

scalaire uniforme se caractérise par :

- l'ensemble des intervalles de quantification $\{[z_i, z_{i+1})\}_{i=0}^{N-1}$ avec $a = z_0 < z_1 < \dots < z_N = b$;
- l'ensemble des valeurs reconstruites $\hat{X}_0, \hat{X}_1, \dots, \hat{X}_{N-1}$, chacune correspondant à son intervalle de quantification.

Les valeurs reconstruite \hat{X}_i sont les espérances des valeurs d'entrée sachant l'intervalle de quantification :

$$\hat{X}_i = \mathbb{E}_{p(x)} [x | x \in [z_i, z_{i+1})] = \frac{\int_{z_i}^{z_{i+1}} xp(x)dx}{\int_{z_i}^{z_{i+1}} p(x)dx}. \quad (1.16)$$

Pour une loi $p(x)$ uniforme, la reconstruction optimale dans le sens de l'erreur quadratique moyenne est donnée par :

$$\hat{X}_i^{UNI} = \frac{z_{i+1} + z_i}{2}. \quad (1.17)$$

Le quantificateur est uniforme si $\forall i : z_{i+1} - z_i = \Delta$, Δ étant appelée le pas de quantification. Ainsi le quantificateur uniforme est défini par un seul paramètre N - le nombre d'intervalles de quantification (une façon alternative est de fixer Δ et déduire N comme $N = \lceil \frac{b-a}{\Delta} \rceil$). Le nombre N_b de bits nécessaires pour représenter l'indice de l'intervalle de quantification, est égal à $\lceil \log_2 N \rceil$.

Le quantificateur à zone morte est une légère modification du quantificateur uniforme. Pour lui, seul l'intervalle qui contient la valeur $x = 0$ ($a < 0 < b$) a une taille différente de Δ , taille de tous les autres intervalles. La taille de l'intervalle contenant $x = 0$ (la zone morte) est un multiple de Δ (souvent 2Δ). Cette modification permet d'améliorer la performance du système quand la distribution de la source est concentrée autour de zéro.

Le quantificateur scalaire uniforme n'est optimal en terme de débit-distorsion (voir la section 1.2.2.2), que pour des sources sans mémoire et avec une loi de distribution uniforme. Cependant, étant facile à exploiter et ayant une complexité faible, il est souvent utilisé dans les systèmes réels, par exemple dans DISCOVER (voir la section 2.3.4.2).

1.2.2.2 Théorie débit-distorsion

La théorie débit-distorsion permet de définir le débit minimal atteignable sous contrainte d'une distorsion de reconstruction donnée.

On va commencer par la définition de la distorsion.

Définition 1.4 La fonction de distorsion entre les deux symboles $x \in \mathcal{A}$ et $\hat{x} \in \hat{\mathcal{A}}$ est une fonction

$$d : \mathcal{A} \times \hat{\mathcal{A}} \longrightarrow \mathbb{R}^+ \quad (1.18)$$

de l'ensemble des paires dans $\mathcal{A} \times \hat{\mathcal{A}}$ sur le demi-axe non-négatif des nombres réels. La distorsion $d(x, \hat{x})$ est une mesure de la distance entre x et \hat{x} .

On va supposer dans la suite que $\mathcal{A} = \hat{\mathcal{A}}$. Voici les deux exemples concrets de la fonction de distorsion :

1. Distorsion (ou distance) de Hamming (utilisée souvent pour les source binaires) définie par :

$$d_H(x, \hat{x}) = \begin{cases} 0 & \text{si } x = \hat{x}, \\ 1 & \text{si } x \neq \hat{x}. \end{cases} \quad (1.19)$$

2. Distorsion quadratique (utilisée souvent pour les source à valeurs continues) définie par :

$$d(x, \hat{x}) = (x - \hat{x})^2 \quad (1.20)$$

Définition 1.5 La distorsion entre les deux séquences de symboles $X \in \mathcal{A}^n$ et $\hat{X} \in \mathcal{A}^n$ est définie par

$$d(X, \hat{X}) = \frac{1}{n} \sum_{i=1}^n d(x_i, \hat{x}_i), \quad (1.21)$$

ou x_i, \hat{x}_i sont des symboles des séquences X et \hat{X} respectivement.

La majuscule D désigne habituellement l'espérance de la distorsion à l'égard de la distribution $p(X)$ de X (pour un système de codage donné) :

$$D = \mathbb{E}_{p(X)} [d(X, \hat{X})] = \sum_{X \in \mathcal{A}^n} p(X) d(X, \hat{X}). \quad (1.22)$$

Définition 1.6 Une paire (R, D) est dite réalisable s'il existe un système de codage de débit R tel que

$$\lim_{n \rightarrow \infty} E_{p(X)} [d(X, \hat{X})] \leq D. \quad (1.23)$$

Définition 1.7 La fermeture de l'ensemble des points (R, D) réalisables est dite la région débit-distorsion de la source X .

Définition 1.8 La fonction débit-distorsion $R(D)$ est la borne inférieure de l'ensemble des débits R tels que la paire (R, D) se trouve dans la région débit-distorsion.

D'après le théorème de Shannon, la fonction débit-distorsion $R(D)$ pour une source X identiquement distribuée et sans mémoire (source *i.i.d.*) est le minimum de l'information mutuelle entre X et \hat{X} pour tous les quantificateurs possibles $p(\hat{X}|X)$ assurant la distorsion inférieure à D :

$$R(D) = \min_{p(\hat{X}|X): \mathbb{E}_{p(X, \hat{X})} [d(X, \hat{X})] \leq D} I(X; \hat{X}). \quad (1.24)$$

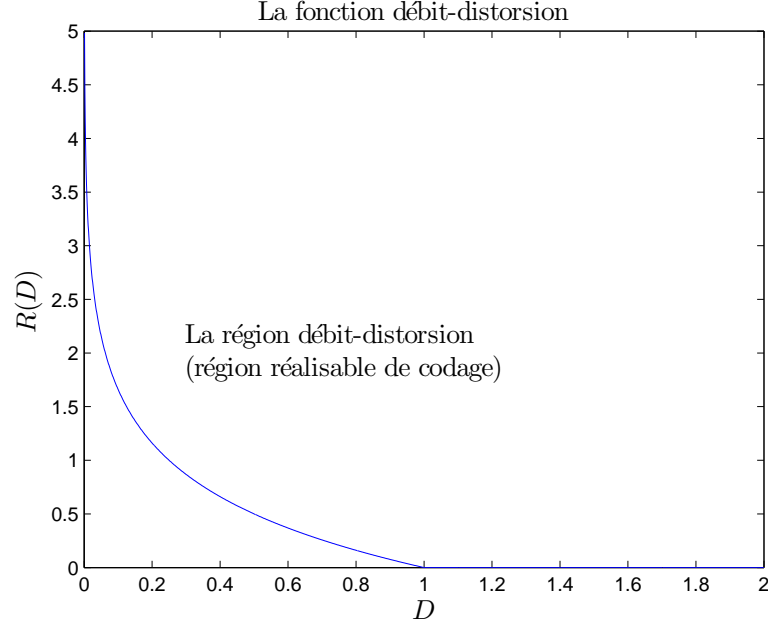


FIG. 1.3 – La fonction débit-distorsion pour une source Gaussienne $\mathcal{N}(0, \sigma^2)$

Pour certaines distributions, il existe une expression analytique pour la fonction $R(D)$ (1.24). Par exemple, la fonction débit-distorsion pour une source Gaussienne $\mathcal{N}(0, \sigma^2)$ de moyenne nulle et de variance σ^2 , et avec la distorsion quadratique, est égale à :

$$R_G(D) = \begin{cases} \frac{1}{2} \log \frac{\sigma^2}{D}, & \text{si } 0 \leq D \leq \sigma^2, \\ 0, & \text{si } D > \sigma^2. \end{cases} \quad (1.25)$$

Cette fonction $R_G(D)$ est représentée sur la figure 1.3.

Dans le cas général, néanmoins, l'obtention de la forme explicite de la fonction $R(D)$ peut être problématique, voire impossible. Cependant il est prouvé (voir chapitre 13 de [CT91], par exemple) que toute fonction débit-distorsion est non-croissante et concave. Cette propriété permet aux systèmes de codage de réaliser l'optimisation de leurs paramètres d'une manière assez efficace décrite dans la section suivante.

1.2.2.3 Optimisation débit-distorsion

Considérons maintenant un système de codage donné, et cherchons la courbe débit-distorsion notée $R(D)$ pour ce système. Ce système est caractérisé par un ensemble de paramètres \mathbf{p} (par exemple, nombre de bande de la transformée, pas de quantification etc).

Il faut trouver un vecteur \mathbf{p}^* de paramètres qui assure le débit minimal $R(\mathbf{p}^*)$ pour

une distorsion $D(\mathbf{p})$ inférieure ou égale à D_{max} :

$$\mathbf{p}^* = \arg \min_{\substack{\mathbf{p} \in \mathcal{P} \\ D(\mathbf{p}) \leq D_{max}}} R(\mathbf{p}), \quad (1.26)$$

où \mathcal{P} est l'ensemble de toutes les configurations possibles du système.

Une manière efficace de réaliser cette optimisation pour différentes contraintes de débit est d'estimer dans un premier temps la fonction débit-distorsion $R(D)$, puis de trouver sur cette fonction un point (D, R) le plus proche de D_{max} du côté inférieur (en se servant du caractère monotone de la fonction débit-distorsion). Pour estimer la fonction $R(D)$ la méthode suivante est souvent utilisée. Un nombre de points p dans \mathcal{P} est parcouru, et ainsi un ensemble de points $(D(p), R(p))$ dans la région débit-distorsion est défini. En utilisant la propriété de concavité de la fonction débit-distorsion, on peut estimer $R(D)$ comme l'enveloppe convexe de cet ensemble de points.

Cette méthode d'estimation de $R(D)$ devient trop complexe si \mathcal{P} est grand, car pour que l'estimation soit assez bonne, le nombre de points parcourus doit représenter une fraction importante de $|\mathcal{P}|$.

En effet, les schémas de codage d'images et de vidéo comprennent très souvent la transformée, qui divise les données en N sous-bandes de fréquence. Ces sous-bandes sont codées indépendamment les unes des autres (donc le nombre de paramètres du système de codage est multiplié par N), et chaque sous-bande contribue sa distorsion D_i et son débit R_i à la distorsion D totale et au débit R total. D'ailleurs, la propriété d'orthogonalité de la transformée et l'hypothèse d'indépendance entre les sous-bandes permettent l'approximation suivante :

$$D(\mathbf{p}) \approx \sum_{i=1}^N D_i(\mathbf{p}_i), \quad (1.27)$$

$$R(\mathbf{p}) \approx \sum_{i=1}^N R_i(\mathbf{p}_i), \quad (1.28)$$

Les fonctions débit-distorsion $R_i(D_i)$ pour chaque sous-bande peuvent habituellement être estimées sans grande difficulté car l'espace \mathcal{P} des paramètres de codage est séparé aussi en N sous-espaces. Par contre, un nouveau problème d'allocation de débit et distorsion entre les sous-bandes est soulevé. Pour résoudre ce type de problème d'optimisation avec une contrainte et à plusieurs paramètres, la méthode de multiplicateurs de Lagrange [Kli04] est souvent utilisée. La fonction de Lagrange pour le problème initial (1.26) est définie par :

$$\Lambda(\mathbf{p}, \lambda) = R(\mathbf{p}) + \lambda(D(\mathbf{p}) - D_{max}), \lambda \in R^+. \quad (1.29)$$

Avec les approximations (1.27)-(1.28), en substituant $R_i(\mathbf{p}_i) = R_i(D_i(\mathbf{p}_i))$, et après un regroupement des membres (1.29) devient :

$$\Lambda(\mathbf{p}, \lambda) = \sum_{i=1}^N [R_i(D_i) + \lambda D_i] - \lambda D_{max}, \lambda \in R^+. \quad (1.30)$$

Son minimum est atteint en $(D_1^*, \dots, D_N^*, \lambda^*)$ lorsque les dérivées correspondantes s'annulent, à savoir :

$$\begin{cases} \frac{\partial R_1}{\partial D_1}(D_1^*) + \lambda^* = 0, \\ \dots \\ \frac{\partial R_N}{\partial D_N}(D_N^*) + \lambda^* = 0, \\ \sum_{i=1}^N D_i^* = D_{max}. \end{cases} \quad (1.31)$$

Le système d'équations (1.31) donne elle-même l'idée d'allocation de débit et de distorsion entre les sous-bandes. En effet, il suffit de trouver l'ensemble de λ tels que toutes les fonctions débit-distorsion $R_i(D_i)$ ont la pente commune $(-\lambda)$ sur les courbes, et puis de trouver dans cet ensemble une valeur λ^* qui correspond à la contrainte $\sum_{i=1}^N D_i^* = D_{max}$. Cette procédure est généralement utilisée dans les codeurs actuels, la recherche de la pente commune et de la valeur λ^* étant souvent réalisée par dichotomie. Nous aussi allons l'utiliser dans la Section 4.5 où nous présenterons une méthode de l'optimisation débit-distorsion pour en codeur vidéo Wyner-Ziv avec une voie de retour.

1.3 Codage de canal

La théorie du codage de canal fut développée par Shannon [Sha48] pour établir les bornes fondamentales de la vitesse de transmission pour le problème de transmission des signaux par des canaux bruités. Par exemple, cette théorie montre que si on veut transmettre les données dans un canal bruité avec une probabilité d'erreur faible, on doit diminuer le taux de transmission en ajoutant de la redondance dans les données.

Le but du codage de canal est alors l'inverse du but du codage de source, car le codage de source diminue la redondance des données. Cependant, cette séparation est justifiée par les théorèmes de la théorie d'information, qui montrent que le codage de source suivi par le codage de canal est optimal (dans le cas où il y a un seul transmetteur et un seul récepteur). Ce principe se comprend intuitivement : le codage de source élimine toute redondance, pendant que le codage de canal en réintroduit une quantité mesurée et dans la manière la plus efficace possible. Un schéma d'un système de communication comprenant le codage de source et le codage de canal, est montré sur la figure 1.4.

Dans cette section nous présentons quelques notions théoriques de base (notamment celle de capacité d'un canal), puis nous décrivons deux techniques de codage de canal les plus utilisées qui seront référencées dans la suite (les codes turbo et les codes LDPC).

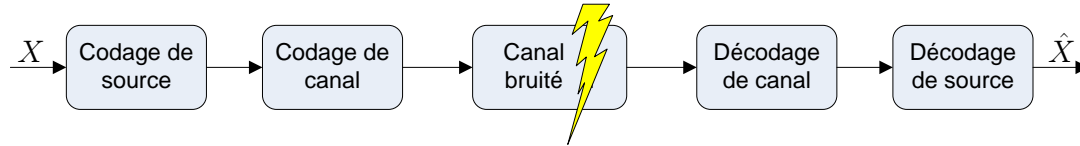


FIG. 1.4 – Schéma générale d'un système de communication

1.3.1 Capacité de canal

On commence par la définition d'un canal discret.

Définition 1.9 Un canal discret est un système qui change le symbole d'entrée $x \in \mathcal{A}$ en un symbole de sortie $y \in \mathcal{B}$, où les alphabets \mathcal{A} et \mathcal{B} sont discrets, avec une probabilité de transition $p(y|x)$, représentant la probabilité d'observer le symbole y en sortie quand le symbole x est entré dans le canal. Le canal est dit sans mémoire si la loi de probabilité $p(y|x)$ ne tient compte que du symbole d'entrée actuel, et est indépendante conditionnellement des entrées et des sorties précédentes du canal.

Autrement dit, un canal est un objet mathématique qui représente formellement le bruit que subit un signal lorsqu'il est transmis par un procédé réel. Introduisons maintenant la notion de capacité de canal, qui donne la limite fondamentale pour la quantité d'information transmissible sur un canal avec une probabilité d'erreur arbitrairement petite.

Définition 1.10 La capacité d'un canal discret sans mémoire est le maximum de l'information mutuelle entre X et Y sur toutes les distributions de X possibles :

$$C = \max_{p(x)} I(X; Y). \quad (1.32)$$

Un exemple de canal discret est un canal binaire symétrique (CBS) caractérisé par la probabilité de *crossover* p (voir figure 1.5(a)). En sortie du canal on observe avec la probabilité $(1 - p)$ les mêmes symboles binaires qu'en entrée, et avec la probabilité p les symboles inversés. La capacité de ce canal est définie par :

$$C = 1 - H(p). \quad (1.33)$$

Un autre type de canal souvent utilisé est un canal à bruit Gaussien additif (en anglais AWGN - Additif White Gaussian Noise channel, présenté schématiquement sur la figure 1.5(b)) dont l'entrée X et la sortie Y sont des sources à valeurs continues :

$$Y = X + Z, \quad Z \sim \mathcal{N}(0, \sigma^2). \quad (1.34)$$

Contrairement au canal binaire symétrique qui sert le plus souvent comme un modèle mathématique simple, le canal Gaussien se trouve souvent en réalité. La capacité

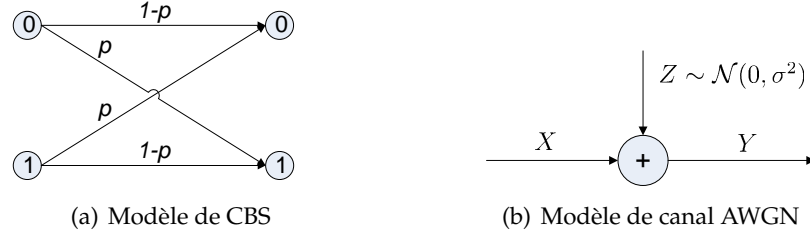


FIG. 1.5 – Modèles des canaux

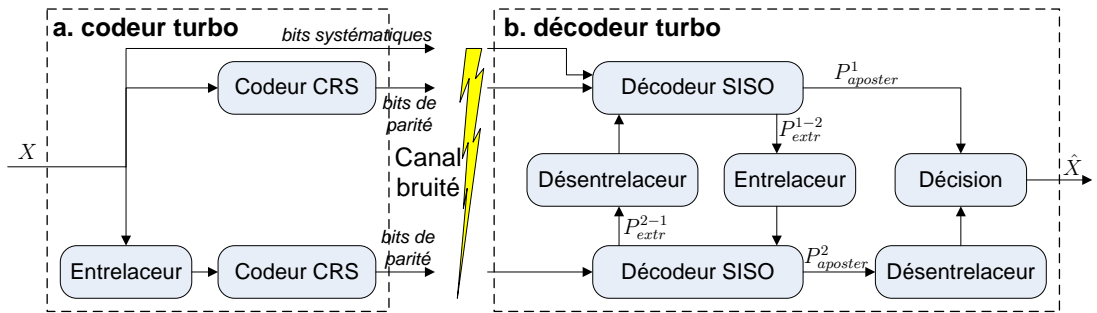


FIG. 1.6 – Schéma d'un codeur turbo

de canal à bruit Gaussien additif est donnée par :

$$C = \frac{1}{2} \log \left(1 + \frac{P}{\sigma^2} \right), \quad (1.35)$$

ou P est une contrainte sur la puissance moyenne du signal d'entrée : $\frac{1}{n} \sum_{i=1}^n x_i^2 \leq P$.

Dans les sections suivantes on va décrire les deux techniques de codage de canal les plus utilisées de nos jours, à savoir les codes turbo et les codes LDPC. Ces deux techniques approchent efficacement la borne fondamentale de la performance (la capacité C). En plus, elles ont une complexité faible, ce qui a beaucoup contribué à leur popularité.

1.3.2 Les codes turbo

Le principe des codes turbo fut découvert par Berrou *et al.* [BGT93, BG96] assez récemment. Ce principe consiste à utiliser deux codes correcteurs d'erreurs convolutifs dont le décodage se fait d'une manière itérative avec la sortie du premier décodeur connectée à l'entrée du deuxième, et réciproquement, d'où une ressemblance avec un moteur turbo, qui utilise lui aussi l'énergie d'échappement pour améliorer sa performance.

Un codeur turbo est présenté sur la figure 1.6a. Il comprend deux codeurs convolutifs récurrents systématiques (CRS) identiques concaténés en parallèle. Notons

que la séquence d'entrée passe par un entrelaceur aléatoire et non-uniforme devant le deuxième codeur pour éviter les cycles dans les graphes de dépendances qui permettent un décodage itératif quasi optimal.

Le décodeur turbo présenté sur la figure 1.6b comprend deux décodeurs souples (en anglais SISO : Soft-In Soft-Out) concaténés en série via l'entrelaceur/désentrelaceur. Pendant le décodage, les deux décodeurs souples s'échangent une information extrinsèque sur les symboles d'entrée, qui est calculée à partir de bits de parité et l'information extrinsèque de l'autre décodeur souple.

Ces itérations s'arrêtent lorsqu'un critère d'arrêt sur les probabilités *a posteriori* des symboles est vérifié, ou après un nombre fixe d'itérations. Le calcul de l'information extrinsèque et de probabilités *a posteriori* dans chaque décodeur souple se fait avec l'algorithme dit BCJR (nommé par les premières lettres des auteurs) [BCJR74], parfois appelé aussi l'algorithme MAP (Maximum A Posteriori) symbole.

Comme nous l'avons déjà écrit, les avantages des codes turbo sont ses bonnes performances (approchant la capacité de canal pour des séquences assez longues) avec une complexité relativement faible. En plus, les codeurs turbo sont très flexibles - avec la technique de poinçonnage le même code convolutif de taux bas fixe (e.g. 1/3) peut être utilisé pour atteindre des taux plus élevés. Les champs d'application de codes turbo incluent la téléphonie mobile (CDMA2000, 3GPP, UMTS), Internet, la télévision etc.

1.3.3 Les codes LDPC

Les codes LDPC (du nom anglais Low-Density Parity Check codes) sont connus depuis longtemps. Ils ont été découverts par Gallager en 1962 [Gal62]. Ces codes approchent la capacité de canal pour les canaux à bruit additif Gaussien.

Un code LDPC est un code en blocs (n, k) décrit par sa matrice de parité \mathbf{H} , c'est-à-dire une séquence d'entrée X de longueur n est transformée en séquence de sortie $Y = \mathbf{H}^T X$ (appelée aussi un syndrome). Cette matrice \mathbf{H} est parcimonieuse (contient très peu de 1 et beaucoup de 0), d'où la "densité faible" dans le nom de ce code. Si le nombre des 1s dans chaque ligne (chaque colonne) est le même, le code est dit régulier, sinon irrégulier.

Le plus souvent le décodage d'un code LDPC se fait par l'algorithme de propagation de croyances, qui calcule itérativement les probabilités *a posteriori* des symboles à la base des probabilités *a priori* et les sommes de parité respectives. Ces probabilités *a posteriori* deviennent les probabilités *a priori* dans l'itération suivante. Pour plus de détails sur le processus de codage et de décodage des codes LDPC, le lecteur peut se référer à la section 2.1.2.5 ou bien à [Mac99].

Grâce à la technique d'accumulation des syndromes proposé par D. Varodayan *et al.* [VAG05], les codes LDPC peuvent être aussi adaptés au débit désiré (la technique de poinçonnage normale comme celle utilisée pour les codes turbo, diminue la perfor-

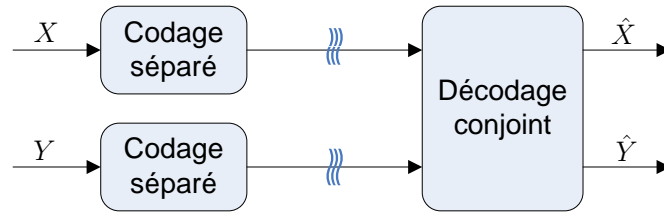


FIG. 1.7 – Exemple d’un système de codage de sources distribuées

mance des codes LDPC d’une manière significative).

Les résultats de comparaison entre les codes turbo et les codes LDPC peuvent être trouvés dans [VAG05, dis07] (application au codage vidéo distribué), [OMK⁺06] (application à la communication). De manière générale, les codes LDPC montrent de meilleures performances que les codes turbo, mais cela dépend du rapport signal à bruit. Le codage/décodage avec les codes LDPC est moins complexe qu’avec les codes turbo [VAG05, OMK⁺06].

1.4 Codage de sources distribuées

Jusqu’ici, nous nous sommes concentrés essentiellement sur le codage d’une source. Le seul résultat que l’on avait évoqué pour plusieurs sources, était la limite de Shannon de taux de compression sans perte pour le codage conjoint de deux sources X et Y qui est égale à l’entropie conjointe $H(X, Y)$. Dans cette section nous allons aborder un nouveau paradigme qui s’appelle le codage de sources distribuées (DSC pour “Distributed Source Coding” en anglais). Ce paradigme n’autorise pas les codeurs à communiquer entre eux (le décodage se fait toujours conjointement). Un exemple de système de codage de sources distribuées est donné sur la figure 1.7.

Les résultats théoriques pour ce paradigme de codage sont obtenus en 1973 par Slepian et Wolf [SW73] pour la compression sans perte, et en 1976 par Wyner et Ziv [WZ76] pour la compression avec pertes. Nous allons présenter des extraits de leurs théorèmes ci-dessous.

1.4.1 Théorème de Slepian-Wolf

Le théorème de Slepian et Wolf [SW73] définit les contraintes sur les débits séparés R_X, R_Y et le débit commun $R_X + R_Y$ pour toutes les configurations possibles de codage de deux sources X et Y . Ces configurations représentées par les états des interrupteurs S_1, S_2, S_3, S_4 sur la figure 1.8 : si l’interrupteur S_i est fermé, la communication est possible, sinon aucune communication n’est possible dans le sens indiqué.

On va s’intéresser aux cas qui correspondent au paradigme du codage de sources distribuées, notamment aux cas où les interrupteurs S_1 et S_2 sont ouverts, pendant

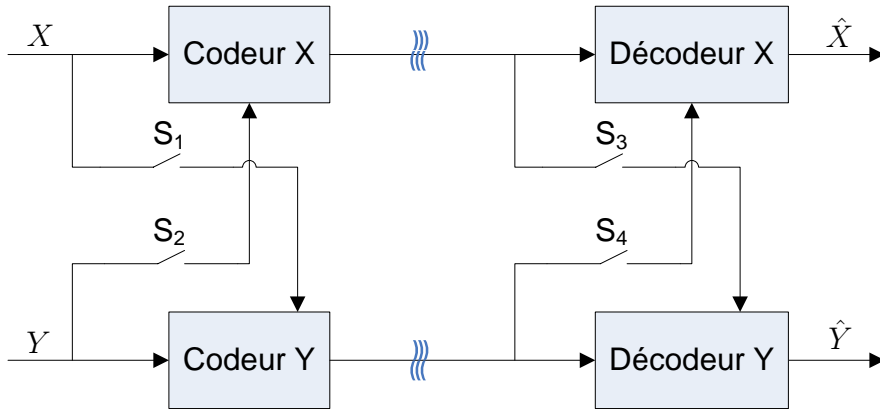


FIG. 1.8 – Illustration pour le théorème de Slepian et Wolf

que les interrupteurs S_3 et S_4 sont fermés tous les deux, ou bien seulement l'un des deux.

Si les deux interrupteurs S_3 et S_4 sont fermés, on parle du cas symétrique de codage de sources distribuées. Dans ce cas-là, les contraintes nécessaires et suffisantes sur les débits pour un décodage réussi (i.e. avec un taux d'erreur par bit quasi-nul) sont les suivantes :

$$R_X \geq H(X|Y), \quad (1.36)$$

$$R_Y \geq H(Y|X), \quad (1.37)$$

$$R_X + R_Y \geq H(X, Y). \quad (1.38)$$

La région de débit réalisable qui correspond aux contraintes (1.36)-(1.38) est présentée sur la figure 1.9. Le résultat important est contenu dans la dernière contrainte (1.38). En effet, elle montre que la limite théorique sur le débit commun de X et Y codés séparément, est celle de Shannon pour le codage conjoint, à savoir $H(X, Y)$. Autrement dit, la communication entre les codeurs de X et Y , qui peut coûter l'énergie ou la complexité dans les systèmes réels, peut être sacrifiée sans aucun impact (théoriquement) sur les performances. Évidemment, c'est la complexité du décodeur qui va augmenter en pratique. Néanmoins, certaines applications peuvent profiter de ce décalage de la complexité du codeur vers le décodeur.

Regardons aussi le cas dit asymétrique, quand l'un des interrupteurs S_3 et S_4 est fermé et l'autre est ouvert. Supposons que S_4 est fermé, i.e. la source X est décodée avec la connaissance de Y . Dans ce cas, la borne AB de débit minimal dans la figure 1.9 se réduit au point A , car les débits nécessaires sont les suivants :

$$R_X \geq H(X|Y), \quad (1.39)$$

$$R_Y \geq H(Y), \quad (1.40)$$

$$R_X + R_Y \geq H(X, Y). \quad (1.41)$$

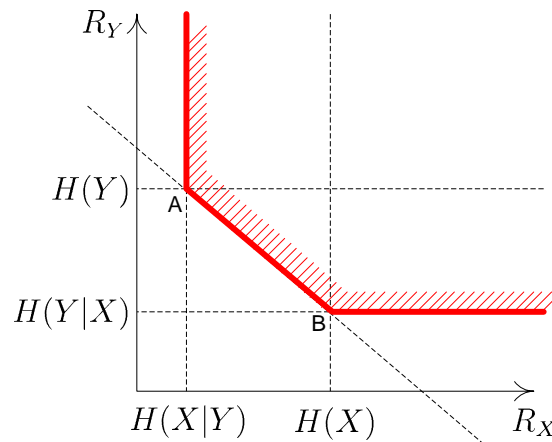


FIG. 1.9 – Région réalisable de débit pour le codage de Slepian-Wolf

Ce cas a une importance pratique et s'appelle le codage distribué avec information adjacente. La source Y qui est compressée à son entropie, s'appelle l'information adjacente. Elle est utilisée pour décoder la source X . Plus la corrélation entre X et Y est élevée, plus le taux de compression de X est important.

La plupart des systèmes réels de codage sont des systèmes de codage avec information adjacente. Les résultats théoriques pour le codage de sources distribuées avec pertes présentés dans la suite se rapportent aussi à ce cas-là dit asymétrique.

1.4.2 Théorème de Wyner-Ziv

Dans cette section nous allons présenter les résultats de la théorie débit-distorsion (voir section 1.2.2.2) liés au codage de sources distribuées avec pertes. Ces résultats furent obtenus par Wyner et Ziv en 1976 [WZ76]. Ils considèrent le schéma de codage présenté sur la figure 1.10. Tout comme dans le cas de la compression sans perte, nous allons distinguer trois cas suivant les états des interrupteurs A et B.

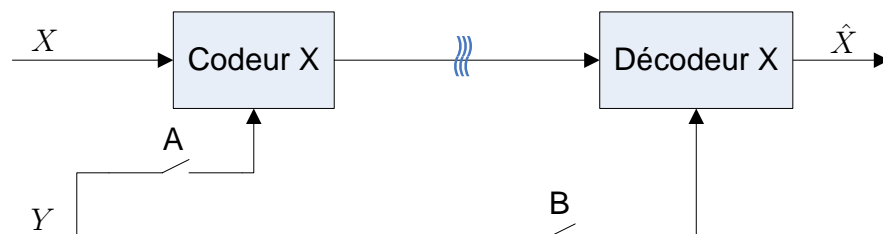


FIG. 1.10 – Illustration pour le théorème de Wyner et Ziv

- i) Les interrupteurs A et B sont ouverts. Le cas du codage séparé de X : Y n'est pas disponible ni au codeur, ni au décodeur. La performance est bornée par la fonc-

tion débit-distorsion $R_X(D)$ classique définie par Shannon (pour une source) :

$$R_X(D) = \min_{p(\hat{X}|X): \mathbb{E}_{p(X, \hat{X})}[d(X, \hat{X})] \leq D} I(X; \hat{X}). \quad (1.42)$$

ii) Les interrupteurs A et B sont fermés.

L'information adjacente est disponible au codeur et au décodeur. Dans ce cas la fonction conditionnelle débit-distorsion est définie par (voir [Ber72] par exemple) :

$$R_{X|Y}(D) = \min_{p(X, Y, \hat{X}): \mathbb{E}_{p(X, Y, \hat{X})}[d(X, \hat{X})] \leq D} I(X; \hat{X}|Y). \quad (1.43)$$

iii) L'interrupteur A est ouvert, B est fermé.

L'information adjacente est disponible seulement au décodeur, mais non pas au codeur, donc le cas asymétrique du codage de sources distribuées. Le théorème de Wyner-Ziv définit la fonction débit-distorsion dans ce cas comme :

$$R_{X|Y}^*(D) = \min_{p(Z, Y, \hat{X}) \in \mathcal{M}(D)} [I(X; Z) - I(Y; Z)], \quad (1.44)$$

où Z est une variable aléatoire telle que Y et Z sont conditionnellement indépendantes de X , et :

$$\sum_z p(x, y, z) = Q(x, y), \quad (1.45)$$

et la minimisation est sur l'ensemble $\mathcal{M}(D)$ de toutes les distributions $p(x, y, z)$ possibles ayant la propriété qu'il existe une fonction de reconstruction $f : Y \times Z \rightarrow \hat{X}$ telle que :

$$\mathbb{E}_{p(X, Y, Z)} [d(X, \hat{X})] \leq D, \text{ avec } \hat{X} = f(Y, Z). \quad (1.46)$$

Les variables X, Y, Z et \hat{X} peuvent être vues comme étant générées par le système de la figure 1.11.

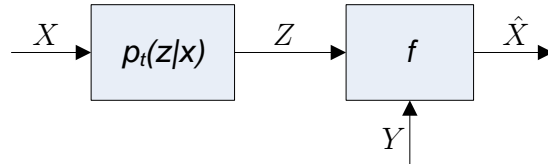


FIG. 1.11 – Illustration du rôle des variables X, Y, Z et \hat{X}

Il est aussi montré dans [WZ76] comme une conséquence de (1.44), que :

$$R_{X|Y}(D) \leq R_{X|Y}^*(D) \leq R_X(D), \quad (1.47)$$

où l'inégalité de droite est naturelle car l'utilisation de l'information adjacente au décodeur ne peut qu'améliorer la performance.

Cependant, il est intéressant de définir les cas où l'inégalité de gauche en (1.47) se transforme en égalité stricte. Premièrement, Wyner et Ziv confirmèrent le résultat de Slepian et Wolf en montrant que $R_{X|Y}^*(0) = R_{X|Y}(0) = H(X|Y)$.

Deuxièmement, ils définirent les conditions suffisantes pour que $R_{X|Y}^*(D) = R_{X|Y}(D)$. Ces conditions, bien que sévères, sont vérifiées, par exemple, pour une variable Gaussienne X , variable Gaussienne $Y = X + U$, où U est aussi une variable aléatoire Gaussienne, et la mesure de distorsion quadratique $d(x, \hat{x}) = (x - \hat{x})^2$. Dans ce cas-là, la fonction débit-distorsion est définie par :

$$R_{X|Y}^*(D) = R_{X|Y}(D) = \begin{cases} \frac{1}{2} \log \frac{\sigma_U^2 \sigma_X^2}{(\sigma_U^2 + \sigma_X^2)D}, & \text{si } 0 < D \leq \frac{\sigma_U^2 \sigma_X^2}{\sigma_U^2 + \sigma_X^2}, \\ 0, & \text{si } D > \frac{\sigma_U^2 \sigma_X^2}{\sigma_U^2 + \sigma_X^2}, \end{cases} \quad (1.48)$$

où σ_X^2 et σ_U^2 sont les variances des variables X et U respectivement. Plus tard, Pradhan *et al.* [PCR03] élargirent ce cas où l'égalité $R_{X|Y}^*(D) = R_{X|Y}(D)$ est validée, en montrant que les variables X et Y peuvent avoir n'importe quelles distributions à condition que U soit Gaussienne.

Dans la théorie du codage de sources distribuées, le résultat de Wyner et Ziv pour la compression avec pertes a la même importance que celui de Slepian et Wolf pour la compression sans perte. Il montre que dans la théorie (bien que sous certaines conditions) il n'y a pas de perte en performance si la communication entre les codeurs est supprimée.

Malgré l'importance théorique de ces résultats, ils ont été "oubliés" pendant quelques décennies, et les premiers systèmes de codage de sources distribuées ne furent proposés qu'à la fin des années 90.

1.5 Conclusion

Dans ce chapitre, nous avons évoqué les éléments de base de la théorie de l'information. Notamment, la théorie du codage de source, la théorie du codage de canal, et la théorie du codage de sources distribuées ont été abordées. En effet, comme nous allons voir dans la suite, beaucoup de ces éléments seront utilisés dans la construction du système de codage distribué de vidéo, dont l'étude et l'amélioration est l'objectif principal de cette thèse.

Dans le chapitre suivant nous essaierons de montrer les étapes principales et résultats importants depuis les premiers systèmes de codage de sources distribuées proposés à la fin des années 90s, jusqu'à la présentation en 2002 d'une première architecture de codage vidéo par Puri et Ramchandran [PR02, PR03b] (appelée PRISM). Puis, nous allons analyser plus soigneusement les systèmes et les algorithmes liés au codage distribué de vidéo qui parurent depuis cette première approche, afin d'établir l'état de l'art dans ce domaine.

Chapitre 2

Codage de sources et de vidéo distribués : état de l'art et résumé des contributions

Le codage de sources distribuées est un paradigme de codage de source assez nouveau. Bien que les résultats théoriques prometteurs furent obtenus au début des années 70, les premières mises en œuvre concrètes ne parurent que vingt-cinq ans plus tard, à la fin des années 90. La plupart d'entre eux se base sur la techniques de codage de canal, mais certains sont également basés sur la technique de codage de source avec des codes à longueur variable.

Le premier système de codage vidéo distribué (CVD, ou DVC en anglais) fut proposé par Puri et Ramchandran [PR02, PR03b] de l'Université de Californie (Berkeley) en 2002 et s'appelait PRISM. Presque en même temps un autre système, qui est connu aujourd'hui sous le nom d'architecture de Stanford, était développé à l'Université de Stanford par un groupe dirigé par Bernd Girod [AZG02]. Ces deux systèmes ont marqué le début des recherches dans le domaine de codage distribué de vidéo.

L'architecture de Stanford fut reprise par plusieurs groupes de recherche afin de l'améliorer et proposer des solutions plus efficaces pour ses différents composants. Notamment, elle fut utilisée comme point de départ pour le projet européen DISCOVER auquel j'ai participé et contribué.

Le but de ce chapitre est de faire un résumé des contributions faites par différents groupes dans le domaine de codage vidéo, ce qui constituera l'état de l'art dans ce domaine. On s'intéressera particulièrement aux recherches relatives aux architectures de Stanford et celle de DISCOVER, auxquelles se rapportent la majorité des contributions de cette thèse. Le résumé de ces contributions sera présenté également. Cependant, on commencera le chapitre par les systèmes de codage dit de Slepian-Wolf de sources discrètes (le codage sans perte), puis on abordera les systèmes de codage dit de Wyner-Ziv de sources Gaussiennes (le codage avec pertes), pour arriver enfin aux

systèmes de codage distribués de vidéo.

2.1 Codage de Slepian-Wolf des sources discrètes

Cette section est consacrée aux techniques de codage de Slepian-Wolf, dont le but est d'effectuer la compression sans perte de deux sources discrètes corrélées d'une façon distribuée. Parmi ces techniques on distingue deux approches différentes. La première est basée sur les codes à longueur variable (donc, sur le codage de source), tandis que la deuxième utilise les codes de canal (les codes turbo entre autres). Notons que la dernière approche est utilisée plus souvent, notamment dans l'architecture de Stanford et PRISM.

2.1.1 Techniques de codage de Slepian-Wolf utilisant les codes de source

Nous allons présenter deux techniques basées sur les codes à longueur variable, notamment celle proposée par Al Jabri et Al-Issa [JAI97], puis la technique de Zhao et Effros [ZE01, ZE03], qui est d'une certaine manière sa généralisation.

2.1.1.1 Technique de codage de Al Jabri et Al-Issa

La technique de codage de Al Jabri et Al-Issa [JAI97] est l'une des premières mises en œuvre des principes de codage de sources distribuées. C'est un système de codage avec information adjacente, dont le schéma est représenté sur la figure 2.1. L'idée générale est donc de coder l'information adjacente Y au taux égal à son entropie $H(Y)$, puis de trouver un code à longueur variable (CLV) pour coder X au taux le plus près possible de son entropie conditionnelle $H(X|Y)$.

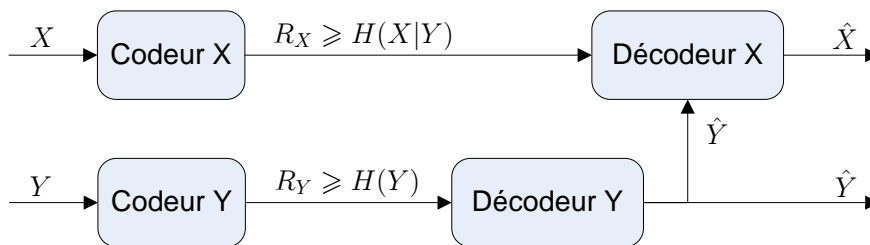


FIG. 2.1 – Codage de Slepian-Wolf avec information adjacente

Illustrons la technique de Al Jabri et Al-Issa par un exemple. Considérons deux sources discrètes X et Y aux alphabets $\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5\}$ et $\mathcal{Y} = \{y_1, y_2, y_3, y_4, y_5\}$

TAB. 2.1 – Les probabilités marginales et entropies de X et Y

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	H
$p_X(x_i)$	4/15	4/15	7/60	7/30	7/60	2.2301
$p_Y(y_i)$	3/10	1/5	3/20	1/10	1/4	2.2282

TAB. 2.2 – Sous-ensembles \mathcal{Y}_i possibles pour chaque $x_i \in \mathcal{X}$

Symbole $x_i \in \mathcal{X}$	Sous-ensemble $\mathcal{Y}_i = \{y_j : y_j \in \mathcal{Y}, p(x_i, y_j) > 0\}$
x_1	$\{y_1, y_2, y_3\}$
x_2	$\{y_1, y_4, y_5\}$
x_3	$\{y_2, y_3\}$
x_4	$\{y_2, y_3, y_4, y_5\}$
x_5	$\{y_4, y_5\}$

respectivement avec la loi de probabilité $p_{xy}(x_i, y_i)$ définie par la matrice suivante :

$$p_{xy}(x_i, y_i) = \begin{pmatrix} 3/20 & 1/15 & 1/20 & 0 & 0 \\ 3/20 & 0 & 0 & 1/30 & 1/12 \\ 0 & 1/15 & 1/20 & 0 & 0 \\ 0 & 1/15 & 1/20 & 1/30 & 1/12 \\ 0 & 0 & 0 & 1/30 & 1/12 \end{pmatrix} \quad (2.1)$$

Les probabilités marginales de X et Y (calculées selon (2.1)) ainsi que leurs entropies sont données au Tab. 2.1.

La source Y peut être compressée séparément avec un taux de compression $R_Y = H(Y) + \epsilon = 2.2282 + \epsilon$, où ϵ est une petite valeur positive (par un code de Huffman, par exemple, voir la section 1.2.1.1). Pour compresser la source X au-dessous de $H(X)$, l'idée de groupement des éléments de \mathcal{X} est exploitée. Par exemple, les symboles x_1 et x_5 peuvent être groupés et codés par un seul mot de code car le décodeur est capable de les décoder sans ambiguïté en utilisant le symbole Y respectif. En effet, les événements $(x_1, y_i), i = 4, 5$, et $(x_5, y_j), j = 1, 2, 3$ sont selon (2.1) des événements de probabilité zéro. Alors, si le décodeur observe y_4 ou y_5 , il décode x_5 , sinon x_1 . La probabilité d'erreur de décodage dans ce cas est nulle.

Pour trouver un groupement optimal, un tableau auxiliaire est créé (Tab. 2.2), contenant pour chaque symbole $x_i \in \mathcal{X}$ le sous-ensemble $\mathcal{Y}_i \subset \mathcal{Y}$ des symboles y_j dont la probabilité conjointe $p(x_i, y_j)$ est positive. En utilisant ce tableau, il est aisé de voir quels symboles de \mathcal{X} peuvent être groupés : il suffit que les sous-ensembles respectifs n'aient pas d'éléments en commun. Dans notre exemple, seuls les groupes suivants sont légitimes : $\{x_1, x_5\}$, $\{x_2, x_3\}$, et $\{x_3, x_5\}$. Par conséquent, il y a cinq partitions de \mathcal{X} possibles, qui sont énumérées dans le Tab. 2.3 avec leurs entropies respectives. L'entropie d'une partition est calculée en regardant un groupe $\{x_i, x_j\}$ comme un seul méta-symbole avec probabilité $(p_X(x_i) + p_X(x_j))$.

TAB. 2.3 – Partitions possibles de X et leurs entropie respectives

Partition	Entropie
$\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}$	2.2301
$\{x_1\}, \{x_2\}, \{x_3, x_5\}, \{x_4\}$	1.9968
$\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}$	1.8903
$\{x_1, x_5\}, \{x_2\}, \{x_3\}, \{x_4\}$	1.8903
$\{x_1, x_5\}, \{x_2, x_3\}, \{x_4\}$	1.5504

Selon le Tab. 2.3, la dernière partition $\{\{x_1, x_5\}, \{x_2, x_3\}, \{x_4\}\}$ a l'entropie la plus intéressante car la plus faible (1.5504). Choissant cette partition et construisant un code de Huffman pour une source X' définie par sa loi de distribution $(\frac{23}{60}, \frac{23}{60}, \frac{7}{30})$, le taux de compression $R_X = H(X') + \epsilon = 1.5504 + \epsilon$ peut être atteint, résultant en taux total $R_X + R_Y = 3.7786 + \epsilon$, ce qui n'est pas très éloigné de la borne théorique $H(X, Y) = 3.6377$.

Cet exemple donne un algorithme assez simple de construction des codes utilisables dans le codage distribué avec l'information adjacente, qui dans certains cas peuvent approcher la borne théorique de codage de Slepian-Wolf, mais il n'est qu'un cas d'école pour comprendre le codage de sources distribuées. La section suivante présente une autre technique, basée également sur les codes à longueur variable.

2.1.1.2 Technique de codage de Zhao et Effros

Il est montré dans [YB00] que la technique de codage de Al Jabri et Al-Issa est sous-optimale. Zhao et Effros [ZE01, ZE03] ont proposé un nouveau formalisme pour la construction des codes optimaux.

Leur schéma de codage est celui de la figure 2.1. On gardera par la suite les mêmes notations, avec deux sources discrètes X et Y d'alphabets finis \mathcal{X} et \mathcal{Y} respectivement. L'idée générale de Zhao et Effros est la même que celle de Al Jabri et Al-Issa - coder l'information adjacente Y à taux égal à son entropie $H(Y)$, puis trouver un code (qui sera optimal dans leur cas) pour le codage de X . Cependant, la partition de X dans leur algorithme est plus compliquée, ayant une structure d'arbre.

Le terme de sous-ensemble est donc remplacé par la notion de groupe de niveau M . Mais avant, il nous faut définir la notion de "confondabilité".

Définition 2.1 Deux symboles $x, x' \in \mathcal{X}$ sont "confondables" (noté par $x \approx x'$) sous $p_{xy}(x, y)$, si $\exists y \in \mathcal{Y} : p(x, y) > 0 \ \& \ p(x', y) > 0$. Dans le cas contraire, ils sont "non-confondables" (noté par $x \not\approx x'$) sous $p_{xy}(x, y)$.

Définition 2.2 L'ensemble $\mathcal{G} = (x_1, \dots, x_m) \subset \mathcal{X}$ est un groupe de niveau 1 sous $p_{xy}(x, y)$, si $\forall x_i, x_j \in \mathcal{G} : (x_i \neq x_j) \Rightarrow x_i \not\approx x_j$. Autrement dit, un groupe de niveau 1 est un sous-ensemble de \mathcal{X} dont tous les éléments sont "non-confondables" par deux.

Voyons maintenant la définition d'un groupe de niveau M .

TAB. 2.4 – La loi de probabilité conjointe $p_{xy}(x, y)$

		y1	y2	y3	y4	y5	y6	y7	y8
$p_{xy}(x_i, y_i)$	x_1	0.04	0	0.04	0.02	0	0	0	0
	x_2	0	0.04	0	0	0.05	0.1	0	0
	x_3	0.15	0	0.05	0	0	0	0	0
	x_4	0	0.05	0	0.06	0	0	0	0
	x_5	0	0.06	0	0	0.05	0	0	0
	x_6	0	0	0	0.01	0.02	0.03	0	0
	x_7	0	0	0.01	0	0	0.06	0.02	0.01
	x_8	0	0	0	0	0	0	0.05	0.08

Définition 2.3 L'ensemble $\mathcal{G} = (\mathcal{R} : C(\mathcal{R}))$ est un groupe de niveau M sous $p_{xy}(x, y)$ avec une racine \mathcal{R} et des descendants $C(\mathcal{R})$, si :

- \mathcal{R} est un groupe de niveau 1,
- $C(\mathcal{R})$ est un ensemble de groupes de niveaux $M - 1$ ou moins (dont au moins un a le niveau $M - 1$),
- chaque paire de symboles $\forall x_r \in \mathcal{R}$ et $\forall x_c \in \mathcal{G}_i, \mathcal{G}_i \in C(\mathcal{R})$ est non-confondable sous $p_{xy}(x, y) : x_r \neq x_c$.

Tous les éléments de \mathcal{R} et des $\mathcal{G}_i \in C(\mathcal{R}), \forall i$, sont considérés comme les éléments du groupe $(\mathcal{R} : C(\mathcal{R}))$ de niveau M , i.e. $x \in (\mathcal{R} : C(\mathcal{R})) := [x \in \mathcal{R} \text{ ou } \exists i : x \in \mathcal{G}_i, \mathcal{G}_i \in C(\mathcal{R})]$.

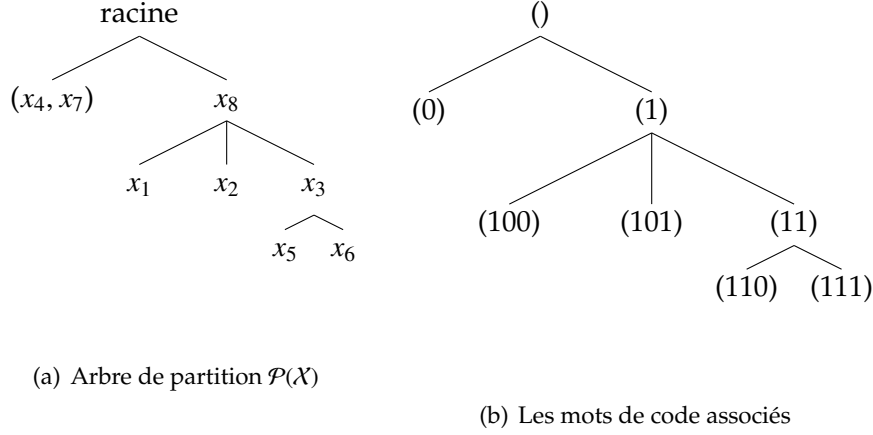
Définition 2.4 L'ensemble complet des groupes $(\mathcal{G}_1, \dots, \mathcal{G}_n)$ non superposés est une partition $\mathcal{P}(\mathcal{X})$ sous $p_{xy}(x, y)$. Autrement dit, $\mathcal{P}(\mathcal{X}) = (\mathcal{G}_1, \dots, \mathcal{G}_n)$ est une partition sous $p_{xy}(x, y)$, si chaque $\mathcal{G}_i, i = 1, \dots, n$, est un groupe de niveau M_i sous $p_{xy}(x, y)$, $\bigcup_{i=1}^n \mathcal{G}_i = \mathcal{X}$, et $\forall i \neq j : \mathcal{G}_i \cap \mathcal{G}_j = \emptyset$.

Prenons un exemple de $p_{xy}(x, y)$ défini par le Tab. 2.4. Pour cet exemple, les groupes légitimes de niveau 1 sont : $(x_1), (x_2, x_3), (x_1, x_5, x_7)$ etc. Un exemple de groupe légitime de niveau 2 est $((x_5) : \{(x_1), (x_3, x_8), (x_7)\})$. En effet, chaque élément du groupe racine (x_5) est non-confondable avec chacun des éléments de l'ensemble $\{x_1, x_3, x_8, x_7\}$. Un exemple de groupe légitime de niveau 3 est $((x_8) : \{(x_1), (x_2), ((x_3) : \{(x_5), (x_6)\})\})$.

Une façon naturelle de représenter un groupe de niveau $M > 1$ est un arbre, dont les branches connectent les descendants à la racine. En effet, toute partition $\mathcal{P}(\mathcal{X}) = (\mathcal{G}_1, \dots, \mathcal{G}_n)$ de \mathcal{X} sous $p_{xy}(x, y)$ peut être représentée comme un arbre, dont la racine est vide, et les descendants sont tous les groupes $(\mathcal{G}_1, \dots, \mathcal{G}_n)$ des différents niveaux. Un exemple d'une partition pour l'exemple de Tab. 2.4 représentée par un arbre est montré sur la figure 2.2(a).

Définition 2.5 Un code est dit associé à la partition $\mathcal{P}(\mathcal{X})$ s'il vérifie les conditions suivantes :

1. tous les éléments d'un groupe de niveau 1 ont le même mot de code associé ;

FIG. 2.2 – Une partition $\mathcal{P}(X)$ et le CLV associé

2. les mots de code associés aux descendants $C(\mathcal{R}) = (\mathcal{G}_1, \dots, \mathcal{G}_n)$ de la racine \mathcal{R} d'un groupe $(\mathcal{R} : C(\mathcal{R}))$ de niveau $M > 1$ ont comme préfixe le mot de code associé à la racine \mathcal{R} ;
3. parmi les mots de code associés aux racines $(\mathcal{R}_1, \dots, \mathcal{R}_n)$ des descendants $C(\mathcal{R}) = (\mathcal{G}_1, \dots, \mathcal{G}_n)$ de la racine \mathcal{R} d'un groupe $(\mathcal{R} : C(\mathcal{R}))$ de niveau $M > 1$, aucun n'est un préfixe d'un autre.

La figure 2.2(b) montre un exemple de code associé à la partition $\mathcal{P}(X)$ montrée sur la figure 2.2(a).

Étant donné une partition $\mathcal{P}(X)$, le code optimal associé (i.e. ayant l'espérance de longueur minimale parmi tous les codes possibles) peut être construit par l'algorithme suivant. Prenons un groupe $(\mathcal{R} : C(\mathcal{R}))$ de niveau $M > 1$, dont la racine \mathcal{R} a déjà un mot de code γ associé, et décrivons l'algorithme permettant de générer les mots de code $(\gamma_1, \dots, \gamma_n)$ pour les racines $(\mathcal{R}_1, \dots, \mathcal{R}_n)$ des groupes-descendants $C(\mathcal{R}) = (\mathcal{G}_1, \dots, \mathcal{G}_n)$. Pour ça, appliquons l'algorithme de Huffman à une source X' avec la loi de probabilité suivante :

$$p_{X'}(x = a_i) = \frac{\sum_{x'' \in \mathcal{G}_i} p_X(x'')}{\sum_{j=1}^n \sum_{x'' \in \mathcal{G}_j} p_X(x'')}, \quad i = 1, \dots, n, \quad (2.2)$$

où $p_X(x)$ est la loi de probabilité marginale de X . Intuitivement, la source X' peut se voir comme une source qui ne prend que des valeurs connectées au même nœud de l'arbre de la partition $\mathcal{P}(X)$ montrée sur la figure 2.2(a), en gardant les probabilités marginales de ces méta-symboles. Les mots de code de Huffman (h_1, \dots, h_n) sont alors préfixés par γ pour obtenir les mots de code finaux pour $(\mathcal{R}_1, \dots, \mathcal{R}_n)$:

$$\gamma_i = \gamma h_i, \quad i = 1, \dots, n. \quad (2.3)$$

En appliquant cet algorithme pour tous les groupes de niveau $M > 1$ à partir de la racine d'arbre (qui a un mot de code vide associé), tous les groupes-feuilles de niveau

1 auront un mot de code associé. Le théorème 3 dans [ZE01, ZE03] prouve que le code construit avec cet algorithme est un code optimal associé à cette partition.

Revenons à notre exemple de la figure 2.2(a). En omettant la première étape, regardons le groupe $((x_8) : \{(x_1), (x_2), ((x_3) : \{(x_5), (x_6)\})\})$ de niveau 3. Pour lui, la loi de probabilité de la source X' est définie par :

$$\left(\frac{p_X(x_1)}{Q}, \frac{p_X(x_2)}{Q}, \frac{p_X(x_3) + p_X(x_5) + p_X(x_6)}{Q} \right) = \left(\frac{0.1}{Q}, \frac{0.19}{Q}, \frac{0.37}{Q} \right), \quad (2.4)$$

$$\text{où } Q = p_X(x_1) + p_X(x_2) + p_X(x_3) + p_X(x_5) + p_X(x_6) = 0.66. \quad (2.5)$$

L'algorithme de Huffman donne les mots de code suivants pour cette source X' : (00), (01), (1). Étant donné que le groupe-racine (x_8) a le mot de code (1) associé, les mots de code pour les groupes (x_1) , (x_2) et (x_3) sont (100), (101) et (11) respectivement. Le code de Huffman entier associé à la partition $\mathcal{P}(X)$ est montré sur la figure 2.2(b).

Jusqu'ici, nous n'avons pas abordé la question de la construction de la partition. Il est aisé de voir que la performance de codage en dépend beaucoup, étant donné que pour chaque partition on utilise un code optimal. C'est en effet un point faible de l'approche de Zhao et Effros. Dans [ZE01, ZE03] ils ne proposent qu'un algorithme exhaustif (bien qu'avec plusieurs améliorations) de recherche d'une partition optimale, qui a une complexité considérable. En revanche, le théorème 1 dans [ZE01, ZE03] permet d'affirmer qu'un code optimal associé à une partition qui est optimale, est un code sans perte instantané optimal pour coder X avec le schéma de codage montré sur la figure 2.1 (i.e. avec information adjacente Y), car pour tout code possible il existe une partition correspondante.

2.1.2 Techniques de codage de Slepian-Wolf utilisant les codes canal

En 1974, Wyner [Wyn74] montre que le codage Slepian-Wolf peut être mis en œuvre avec des codes canal. En effet, la corrélation entre X et Y peut se voir comme un canal virtuel dit "canal de corrélation" avec une probabilité de transition $p(y|x)$ ou $p(x|y)$. Dans le schéma de codage distribué avec l'information adjacente Y , Y est considéré comme la version bruitée de X avec le "canal virtuel de corrélation" défini par $p(x|y)$. Le but du codage de Slepian-Wolf est donc de reconstituer le signal original X en observant sa version bruitée Y , ce qui est aussi le but du codage de canal.

Nous allons présenter trois approches différentes. La première basée sur les codes blocs, est démonstrative, elle permet de comprendre l'idée générale d'utilisation des codes canal tout comme l'approche de Al Jabri et Al-Issa nous a permis de comprendre l'utilisation des codes source dans le cadre du codage distribué. Les deux approches suivantes sont basées sur les codes turbo et les codes LDPC respectivement.

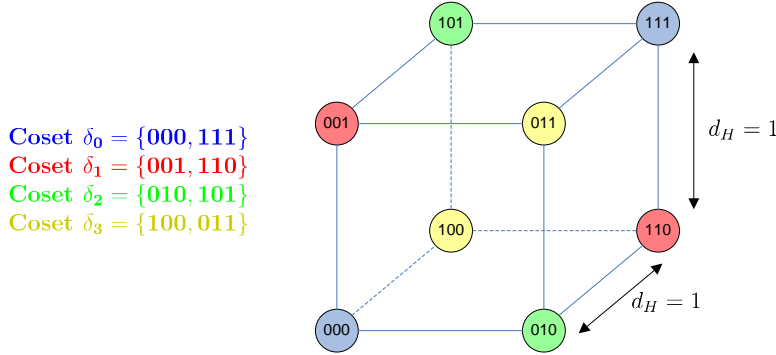


FIG. 2.3 – Illustration pour l'exemple de codage avec DISCUS

2.1.2.1 Codage distribué avec des syndromes - DISCUS

Dans leur approche nommée “codage de sources distribuées en utilisant des syndromes” (en anglais DIs tributed Source Coding Using Syndromes, DISCUS), Pradhan et Ramchandran [PR99] ont repris l'idée de Wyner [Wyn74]. Montrons cette idée par un exemple simple.

Considérons deux sources discrètes X et Y dont les symboles sont des mots constitués de 3 symboles binaires, par exemple 001, 011, 111. Supposons ces symboles équiprobables, i.e. $p_X(x) = p_Y(y) = 0.125$. Par conséquent, $H(X) = H(Y) = 3$ bits.

Supposons aussi que la loi de corrélation entre X et Y est telle que la distance de Hamming (1.19) entre les symboles x et y n'est jamais supérieure à 1 : $d_H(x, y) \leq 1$. Dans ce cas, si le décodeur connaît y , pour décoder x il a besoin de 2 bits d'information seulement, car la connaissance de y et de la loi de corrélation ne laisse que 4 choix possibles (et équiprobables) pour x . Alors, $H(X|Y) = 2$. L'algorithme de Pradhan et Ramchandran permettant d'atteindre la borne de Slepian et Wolf, i.e. de décrire x avec seulement 2 bits (en supposant que le codeur ne connaît pas y), est décrite ci-dessous.

Répartissons les 8 symboles que X peut prendre, en 4 sous-ensembles de deux symboles chacun, d'une telle façon que la distance de Hamming entre les deux symboles d'un sous-ensemble soit maximale (c'est-à-dire égale à 3). Ces sous-ensembles sont : $\delta_0 = \{000, 111\}$, $\delta_1 = \{001, 110\}$, $\delta_2 = \{010, 101\}$, $\delta_3 = \{100, 011\}$, ils sont représentés par des couleurs différentes sur la figure 2.3. Chaque sous-ensemble δ_i (appelé *coset*) peut être représenté par son indice i , qui peut être codé avec 2 bits. Nous montrons dans un premier temps que la connaissance de cet indice suffit au décodeur pour reconstruire X s'il connaît Y , pour conclure que le taux de codage correspondant à la borne de Slepian-Wolf $H(X, Y) = H(Y) + H(X|Y) = 3 + 2 = 5$ peut être atteint.

En effet, soient par exemple $X = 010$, et $Y = 110$. Le codeur, qui n'a pas accès à Y , envoie l'indice 2 du coset δ_2 auquel X appartient. Le décodeur, qui connaît $Y = 110$, reçoit l'indice du coset, et doit choisir entre les deux valeurs de X possibles : 010 et 101. Seulement l'une d'entre elles vérifie la contrainte imposée par la loi de corrélation, à

savoir : $d_H(010, 011) = 1$, $d_H(101, 011) = 2 \geq 1$. Le décodeur décode ainsi $X = 010$ et confirme que le débit total nécessaire pour décoder X et Y , est 5 bits. Le lecteur peut vérifier que la technique fonctionne pour toute autre paire de X et Y .

L'algorithme ci-dessus peut être représenté dans la forme d'un code en blocs (n, k, d) , où n est la longueur du code, k est la longueur du message, et d est la distance minimale entre deux mots de code. Ce code est caractérisé par sa matrice de parité \mathbf{H} , souvent représentée sous la forme systématique : $\mathbf{H} = [\mathbf{A}|\mathbf{I}]$. L'indice i de coset δ_i ici s'appelle *syndrome*, et est donné par la multiplication de la matrice de parité par x : $\mathbf{s} = \mathbf{H}x$.

Le décodeur connaissant \mathbf{s} et y , calcule $y' = y + z$ (l'addition se fait modulo-2), où $z = [0|\mathbf{s}]$ est un mot de code dont le syndrome $\mathbf{H}z = [\mathbf{A}|\mathbf{I}] \times [0|\mathbf{s}]$ est \mathbf{s} . Si x' est le mot de code le plus près de y' (en terme de distance de Hamming), et dont le syndrome est nul, alors la valeur de x peut être déduite comme $x = x' + z$ (à condition que la distance minimale d du code permet de corriger un tel nombre d'erreurs, à savoir si $d_H(x, y) \leq \frac{d-1}{2}$). Le taux de compression dans ce cas est $\frac{n}{n-k}$, avec $H(X|Y) = n - k$.

Pour l'exemple que l'on a étudié ci-dessus, le code $(3, 1, 3)$ est défini par sa matrice :

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad (2.6)$$

En effet, on peut vérifier que les syndromes \mathbf{s} générés par cette matrice \mathbf{H} sont identiques aux indices de cosets.

2.1.2.2 Technique de codage utilisant deux codes turbo poinçonnés

Les codes turbo inventés par Berrou *et al.* [BGT93, BG96] offrent de très bonnes performances en codage de canal. On les utilisera à de nombreuses reprises pour le codage de sources distribuées, traité comme le problème de codage de canal. La première mise en œuvre que l'on va décrire, est celle de Garcia-Frias et Zhao [GFZ01].

Cette mise en œuvre considère deux sources binaires X et Y corrélées (avec une probabilité de transition $p = \Pr(X \neq Y)$), qui doivent être compressées séparément et décodées conjointement. Les auteurs proposent de coder chaque source avec un codeur turbo dont les sorties (les bits $c_k^{t,s}$ sur la figure 2.4) sont poinçonnées (les bits systématiques comme les bits de parité). L'idée de base est que chacun des deux codeurs turbo sort une moitié des bits systématiques, tandis que le nombre des bits de parité est ajusté au débit total désiré. C'est alors un schéma symétrique de codage de sources distribuées.

Le décodage se fait par deux décodeurs turbo connectés en série, qui s'échangent de l'information extrinsèque après chaque itération interne. On peut dire que c'est l'idée "turbo", appliqué à son tour aux codeurs turbo (voir figure 2.4). L'information extrinsèque que les décodeurs s'échangent, tient compte de la probabilité p de transition entre X et Y . Il est montré que cette probabilité peut être estimée pendant le décodage sans impact important sur la performance.

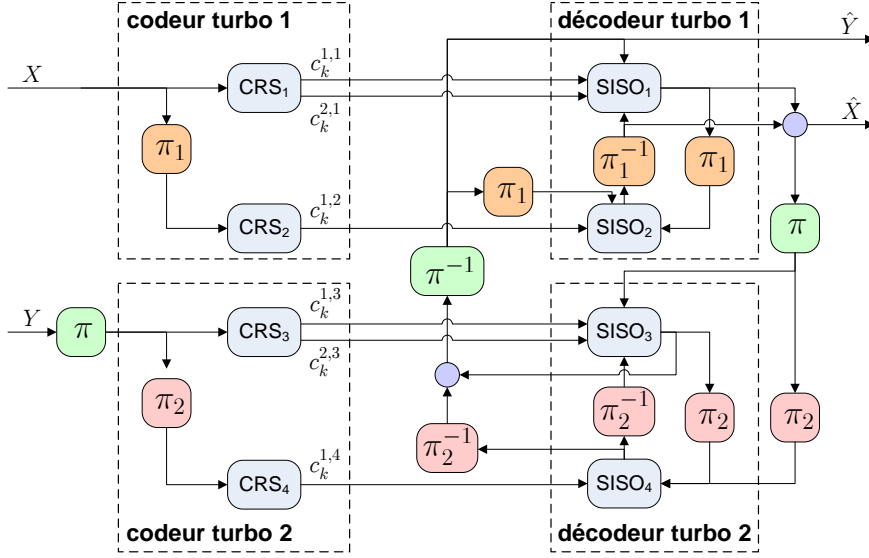


FIG. 2.4 – Schéma de codage de Slepian-Wolf par Garcia-Frias et Zhao

Les résultats des simulations montrent que les performances du système sont dans l'écart d'un facteur de 1.12 de la borne de Slepian-Wolf $H(X, Y)$ (avec le taux d'erreurs moyen au-dessous de 10^{-4}) pour toute une gamme de p entre 0.025 et 0.2. Par conséquent, le système n'est pas capable de compresser les sources avec une probabilité de transition $p \gtrsim 0.25$ (ce qui correspond à peu près à l'entropie conjointe $H(X, Y) \approx \frac{2}{1.12}$).

2.1.2.3 Technique de codage avec des machines à nombre d'états fini

Une autre mise en œuvre du principe turbo pour le codage de Slepian-Wolf fut proposée par Bajcsy et Mitran [BM01]. Dans leur cas, ils considèrent le schéma de codage avec l'information adjacente. Plus précisément l'information adjacente Y est compressée à un taux proche de son entropie $H(Y)$, tandis que le taux de compression de X recherché est autour de son entropie conditionnelle $H(X|Y)$.

Contrairement à l'approche de Garcia-Frias et Zhao, Bajcsy et Mitran utilisent deux machines à nombre d'états fini (en anglais Finite-State Machines, ou FSM) identiques aux codeurs élémentaires constituant le codeur turbo, et non pas des codes convolutionnels. Chaque FSM pour k bits d'entrée donne n bits en sortie, $k > n$; en conséquence, le taux de compression de chaque FSM est $\frac{k}{n}$, et le taux total de X est $\frac{k}{2n}$. Un exemple de FSM pour $k = 2$ et $n = 1$ est montré sur la figure 2.5. Sur les branches les entrées et les sorties de FSM selon son état actuel sont marquées.

Le décodage se fait avec un algorithme BCJR [BCJR74], appliqué au treillis du codeur FSM : chaque décodeur estime les probabilités *a posteriori* en utilisant la sortie du codeur FSM respectif, Y , et l'information extrinsèque reçue de l'autre décodeur. L'information extrinsèque est modélisée comme un vecteur des réalisations d'une va-

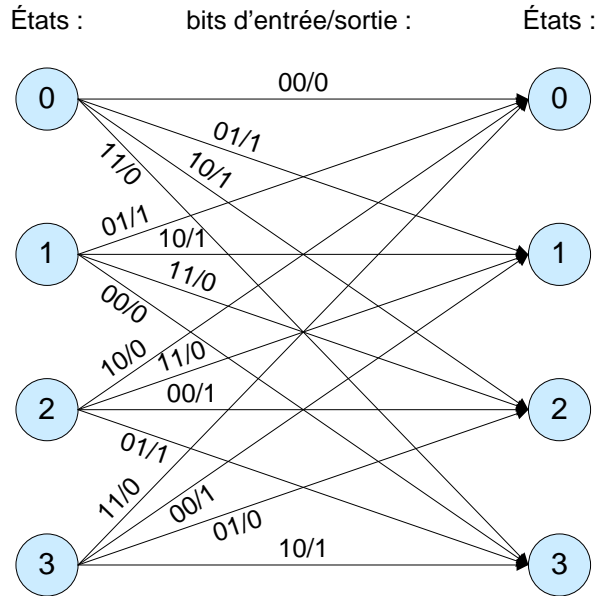


FIG. 2.5 – Exemple d’une machine à nombre d’états fini (4 états)

riable aléatoire Gaussienne sans mémoire avec une espérance de 0.5 et une variance de 1 (étant donné que X et Y sont binaires).

Les performances du système de codage de Bajcsy et Mitran sont comparables à celle du système de Garcia-Frias et Zhao : ils assurent un taux d’erreur $< 10^{-4}$ pour deux sources X et Y (ayant les probabilités marginales de 0.5 et l’entropie conditionnelle $H(X|Y) = 0.49$), codées à un taux total de $1 + 2/3 = 1.67$, ce qui représente un écart de 1.12 par rapport à la borne théorique.

2.1.2.4 Technique de codage avec l’information adjacente en utilisant des codes turbo

La technique de codage de Slepian-Wolf proposée par Aaron et Girod en 2002 [AG02] est dans un certain sens un mélange des deux techniques présentées précédemment. En effet, l’architecture utilisée par Aaron et Girod est celle de Bajcsy et Mitran, à savoir l’architecture avec l’information adjacente Y , où le décodeur de X utilise le principe turbo. En même temps, les codeurs constituant d’un codeur turbo sont des codeurs convolutionnels récurrents systématiques (CRS) comme dans le cas de Garcia-Frias et Zhao, et non pas des machines à nombre d’états fini, ce qui permet au système de s’adapter au débit désiré par la technique de poinçonnage. Le schéma de codage/décodage proposé par Aaron et Girod est montré sur la figure 2.6.

Chacun des deux CRS fonctionne à taux $\frac{n-1}{n}$, c’est-à-dire pour chaque $n - 1$ bits d’entrée il donne n bits en sortie, dont $n - 1$ bits systématiques plus 1 bit de parité. Contrairement à la technique de Garcia-Frias et Zhao, tous les bits systématiques sont

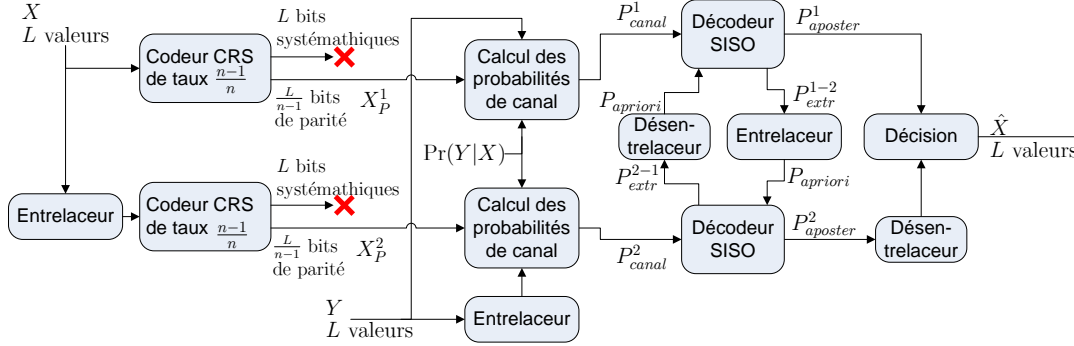


FIG. 2.6 – Schéma de codage de Slepian-Wolf par Aaron et Girod

jetés car il est supposé que le décodeur les a déjà sous une forme bruitée (ce sont les bits de l'information adjacente Y). Par conséquent, le taux de compression de X est $\frac{2}{n-1}$, ou 2 bits de parité par $n - 1$ bits d'information. Si nécessaire, ce taux peut être encore diminué par un poinçonnage des bits de parité.

Le décodage se fait toujours par l'algorithme BCJR, avec la seule différence par rapport au cas de décodage de canal, que les probabilités de canal $p(x_i)$ (qui sont les probabilités *a priori* pour chaque décodeur SISO) sont remplacées par les probabilités conditionnelles $p(x_i|y_i, x_{p,i}^k)$, où $x_{p,i}^k$ sont les bits de parité venant du codeur respectif k .

Avec les résultats des simulations présentés dans [AG02], les auteurs prétendent que leur schéma est plus performant que celui de Garcia-Frias et Zhao. Pour expliquer ce fait, ils font une hypothèse qu'en général une architecture asymétrique est préférable car il existe déjà des techniques très efficaces pour compresser Y à un taux R_Y très proche de $H(Y)$. Ainsi la sous-optimalité du taux total $R_X + R_Y$ est due seulement à la sous-optimalité du taux R_X de codage de X , et non pas à la sous-optimalité de R_Y . La sous-optimalité de leur système selon les résultats des simulations est entre 3 et 11%.

Cette technique de codage de Slepian-Wolf est à la base de l'architecture de Stanford de codage distribué de vidéo, que l'on abordera dans la section 2.3.2.

2.1.2.5 Technique de codage avec l'information adjacente en utilisant des codes LDPC

Cependant, les meilleures performances dans le domaine de codage de Slepian-Wolf au jour d'aujourd'hui ont été obtenues avec les codes LDPC [LXG02, TGFZ03, VAG05].

Les codes LDPC sont les codes en blocs présentés dans la section 2.1.2.1. Leur particularité (un nombre faible de 1s dans la matrice de parité \mathbf{H}) permet un décodage rapide et efficace par un algorithme dit de "propagation des croyances" (appelé aussi l'algorithme de passage des messages), bien que cet algorithme soit sous-optimal.

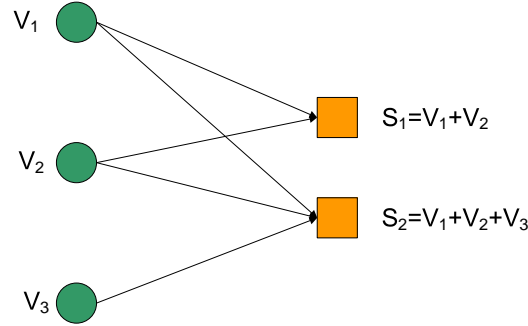


FIG. 2.7 – Graphe “bipartite” du code LDPC

Avant d’expliquer l’algorithme, regardons un exemple simple. Soit la matrice de parité \mathbf{H} d’un code (3,2) :

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad (2.7)$$

Le syndrome \mathbf{s} de longueur 2 bits est $\mathbf{s} = \mathbf{H}^T \mathbf{X}$, où \mathbf{X} est la séquence d’entrée (de 3 bits). Un graphe dit “bipartite” correspondant à \mathbf{H} est montré sur la figure 2.7. Dans ce graphe, les nœuds à gauche (ayant la forme des ronds) représentent les bits d’information (les bits de \mathbf{X}), tandis que les nœuds à droite (ayant la forme des carrés) représentent les bits du syndrome. Les connexions entre les deux types de nœuds correspondent aux 1s dans la matrice \mathbf{H} : chaque bit du syndrome \mathbf{s} est calculé comme une somme modulo-2 des bits d’information connectés à son nœud.

Le décodage par l’algorithme de passage des messages consiste en la transmission des “messages” entre les nœuds d’information et les nœuds de parité (voir la figure 2.8). Ces messages représentent les “croyances” actuelles sur les valeurs des bits d’information ou les bits de parité (cela dépend de la direction de passage du message). En combinant dans chaque nœud les croyances venant des nœuds différents de la partie adverse du graphe, les croyances se mettent à jour, et une nouvelle itération de l’algorithme peut être lancée. Les itérations s’arrêtent lorsque le nombre d’équations de parité non-vérifiées pour la séquence décodée $\hat{\mathbf{X}}$ est nul.

Le message à passer du nœud d’information m au nœud de parité n (s’ils sont connectés dans le graphe bipartite) est calculé avec la fonction Ψ définie par :

$$\Psi_{m,n} = \log \frac{\Pr(y_m | x_m = 0)}{\Pr(y_m | x_m = 1)} + \sum_{i \neq n} q_{i,m}, \quad (2.8)$$

où $q_{i,m}$ sont les messages venant (à l’itération précédente) des nœuds de parité connectés au nœud d’information m , et $\Pr(y_m | x_m = b)$ est la probabilité d’observer y_m quand $x_m = b$. Si la corrélation entre X et Y est modélisée comme un canal binaire symétrique

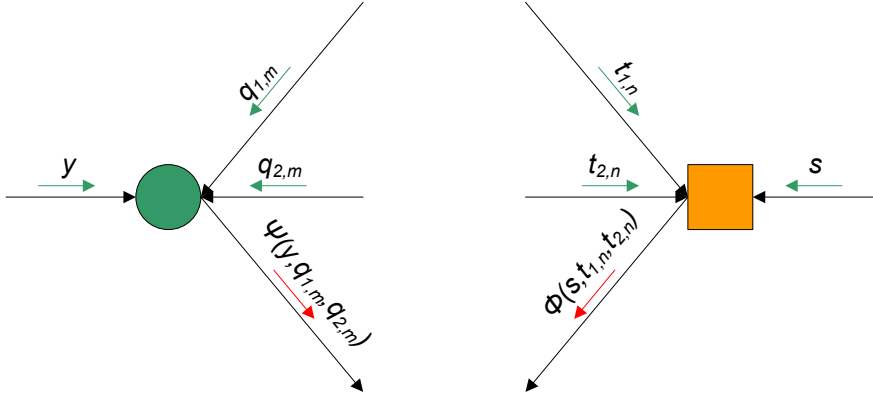


FIG. 2.8 – Algorithme de passage des messages

virtuel avec une probabilité de transition p :

$$\Pr(y|x = b) = \begin{cases} p, & \text{si } y = 1 - b, \\ 1 - p, & \text{si } y = b. \end{cases} \quad (2.9)$$

Dans le sens inverse, c'est-à-dire dans les nœuds de parité, le nouveau message à transmettre depuis le nœud de parité n vers le nœud d'information m , est calculé avec la fonction Φ définie par :

$$\tanh \frac{\Phi_{n,m}}{2} = (1 - 2s) \prod_{j \neq m} \tanh \frac{t_{j,n}}{2}, \quad (2.10)$$

où $t_{j,n}$ sont les messages venant des nœuds d'information connectés au nœud de parité n , et s est le syndrome disponible au décodeur.

Ainsi, l'itération suivante se fait avec :

$$q_{n,m} = \Phi_{n,m}, \quad (2.11)$$

$$t_{m,n} = \Psi_{m,n}, \quad (2.12)$$

pour toutes les branches (m, n) dans le graphe bipartite. Notons que grâce aux conditions $i \neq n$ et $j \neq m$ dans (2.8) et (2.10) respectivement, l'algorithme ne rend pas l'information reçue d'un nœud au même nœud à l'itération suivante. Pour la première itération, les messages sont nuls.

Après l'arrêt de l'algorithme, les valeurs des X sont données par :

$$\hat{x}_m = \begin{cases} 0, & \text{si } \log \frac{\Pr(y_m|x_m = 0)}{\Pr(y_m|x_m = 1)} + \sum_i q_{i,m} \geq 0, \\ 1, & \text{sinon.} \end{cases} \quad (2.13)$$

Liveris *et al.* [LXG02] comparent les performances des codes LDPC avec celles des codes turbo dans les mêmes conditions, et trouvent que les codes LDPC permettent de

se rapprocher encore plus de la borne théorique de Slepian-Wolf, bien qu'ils avouent que l'utilisation des codes LDPC irréguliers (au lieu des codes réguliers) soit nécessaire pour les taux de compression élevés. Cette même conclusion est confirmée en général par les résultats présentés dans [VAG05].

2.2 Codage de Wyner-Ziv des sources Gaussiennes

Dans cette section nous allons présenter des schémas de codage distribué avec pertes, dits les schémas de codage Wyner-Ziv. La plupart de ces schémas considèrent deux sources Gaussiennes X et Y , avec $Y = X + Z$, X et Z étant indépendantes, centrées et de variances σ_X^2 et σ_Z^2 .

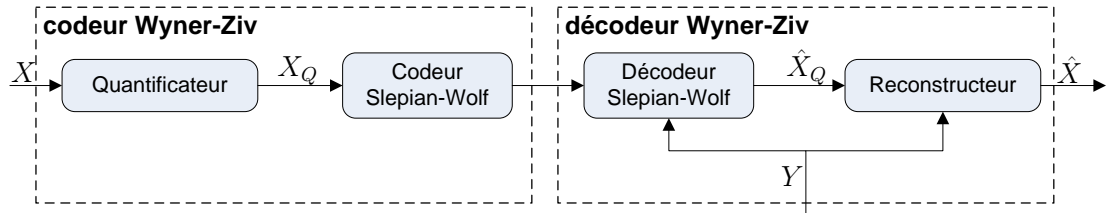


FIG. 2.9 – Schéma d'un codeur de Wyner-Ziv

Le codage avec pertes suppose la présence d'un quantificateur. En effet, le schéma de codage de Wyner-Ziv peut se voir comme le schéma de codage de Slepian-Wolf précédé d'un quantificateur (voir la figure 2.9). Dans le codage distribué, la paire quantificateur-reconstructeur doit être construite de telle manière qu'elle puisse utiliser la corrélation entre X et Y pendant le décodage, mais non pas pendant le codage. On peut classer les schémas existants de codage de Wyner-Ziv en trois grandes classes :

- des schémas avec le quantificateur en lattice emboîtée (par exemple [ZSE02, Ser07, XLCL03]) ;
- des schémas avec le quantificateur combiné avec le codeur de canal (par exemple DISCUS [PR03a]) ;
- des schémas avec le quantificateur standard suivi par un codeur de Slepian-Wolf (par exemple [AG02]).

Nous allons détailler ces trois classes ci-dessous.

2.2.1 Techniques de codage avec le quantificateur en lattice emboîtée

Nous allons commencer cette sous-section par le simple exemple d'un codeur de Wyner-Ziv basé sur le quantificateur en lattice emboîtée. Ensuite nous présenterons un schéma plus avancé qui permet d'approcher la borne de Wyner-Ziv en théorie.

Nous citerons aussi quelques résultats pratiques obtenus avec cette approche (avec un nombre faible de dimensions).

2.2.1.1 Exemple simple

Dans cet exemple nous supposons que X et Y sont deux suites de variables i.i.d. Gaussiennes avec une densité de probabilité jointe $p_{XY}(x, y)$. Supposons aussi que Y est disponible au décodeur. Pour effectuer le codage de Wyner-Ziv X , la méthode suivante peut être employée.

Le vecteur X est quantifié avec un quantificateur scalaire uniforme à 8 niveaux avec un pas de quantification égal à q (notons qu'il peut être vu comme 2 quantificateurs emboîtés, l'un avec un pas q , et l'autre avec un pas $4q$, dont les niveaux sont marqués avec la même couleur sur la figure 2.10). Ces 8 niveaux sont ensuite répartis en 4 sous-ensembles de telle manière que les 2 éléments de chaque sous-ensemble soient les plus éloignés possible l'un de l'autre (on peut remarquer ici une ressemblance avec l'exemple que l'on a présenté pour introduire en codage de sources distribuées avec les codes de canal dans la section 2.1.2.1). Ces 4 sous-ensembles sont marqués avec les 4 différentes couleurs sur la figure 2.10.

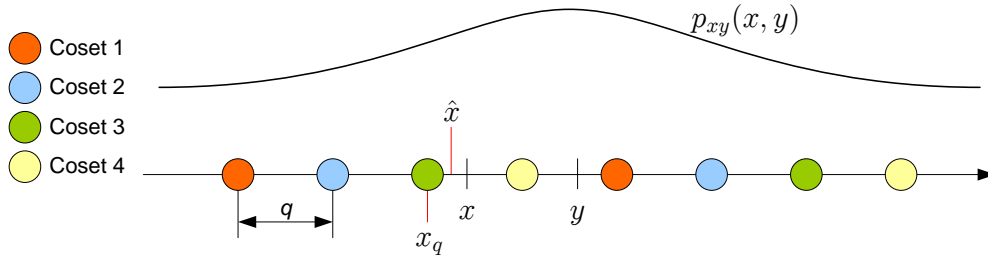


FIG. 2.10 – Illustration pour l'exemple du codage de Wyner-Ziv

Comme dans le cas du codage de Slepian-Wolf, le codeur envoie seulement l'indice du sous-ensemble (nommé coset), codé (dans notre exemple) avec 2 bits. En règle générale, l'indice peut être aussi codé avec un code de source.

Le décodeur ayant reçu le coset, choisit parmi ses éléments (niveaux de quantification) celui qui est le plus proche de la valeur de y . Supposons qu'il s'agisse de l'intervalle $[z_i, z_{i+1})$. Après cela, la valeur \hat{x} se reconstruit comme l'espérance de x sachant l'intervalle de quantification $[z_i, z_{i+1})$ et l'information adjacente y (voir la section 1.2.2.1 pour plus de détails) :

$$\hat{x} = \mathbb{E}_{p_{xy}(x,y)} [x|x \in [z_i, z_{i+1}), Y = y] = \frac{\int_{z_i}^{z_{i+1}} x p_{xy}(x, y) dx}{\int_{z_i}^{z_{i+1}} p_{xy}(x, y) dx}. \quad (2.14)$$

2.2.1.2 Technique de codage de Zamir *et al.*

Une technique de codage plus avancée que l'exemple ci-dessus, fut proposé par Zamir *et al.* [ZS98, ZSE02]. Avant de la présenter, introduisons quelques notions pratiques.

Soit $\Lambda \subset R^n$ une lattice dans l'espace R^n , $n \geq 1$, avec une fonction de reconstruction :

$$Q_\Lambda(\mathbf{x}) = \arg \min_{\ell \in \Lambda} \|\mathbf{x} - \ell\|. \quad (2.15)$$

Les régions de Voronoï $\mathcal{V}_\Lambda(\ell)$ de Λ sont les ensembles définis par :

$$\mathcal{V}_\Lambda(\ell) = \{\mathbf{x} \in R^n : Q_\Lambda(\mathbf{x}) = \ell\}, \quad \ell \in \Lambda. \quad (2.16)$$

Soit $\Lambda' \subset R^n$ une autre lattice. On dit que Λ' est emboîtée en Λ , si $\Lambda' \subset \Lambda$. Intuitivement, Λ' est une lattice de quantification plus grossière que Λ , comme l'illustre la figure 2.11.

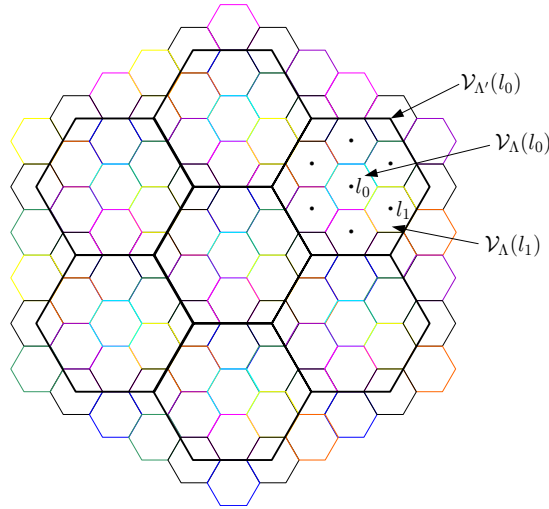


FIG. 2.11 – Exemple d'une paire de lattices emboîtées

Expliquons maintenant le schéma de codage de Zamir *et al.* Deux lattices emboîtées Λ et Λ' sont choisies, et modélisent en quelque sorte le codage de source et le codage de canal respectivement, comme présenté sur la figure 2.9. Elles sont choisies selon deux critères :

- la lattice Λ est assez fine pour permettre la distorsion voulue D :

$$\mathbb{E}_{p_{\mathbf{xy}}} [\|\mathbf{x} - Q_\Lambda(\mathbf{x})\|^2] \leq D; \quad (2.17)$$

- la lattice Λ' est assez grossière pour que la probabilité que l'erreur $\mathbf{Z} = \mathbf{Y} - \mathbf{X}$ dépasse la région 0 de Voronoï $\mathcal{V}_{\Lambda'}(0)$ soit petite :

$$P_e = \Pr\{\mathbf{Z} \notin \mathcal{V}_{\Lambda'}(0)\} < \epsilon. \quad (2.18)$$

Ces conditions impliquent que les régions de Voronoï des lattices Λ et Λ' doivent approcher les sphères euclidiennes de rayon \sqrt{nD} et $\sqrt{n\sigma_Z^2}$ respectivement [ZSE02].

Ensuite, soit \mathbf{u} un vecteur aléatoire distribué uniformément dans $\mathcal{V}_{\Lambda}(0)$, qui est également disponible au décodeur (en utilisant le même générateur pseudo aléatoire, par exemple). Ce vecteur est appelé “dither” en anglais, et son rôle est d’assurer une répartition uniforme des vecteurs de données dans la région de Voronoï correspondante $\mathcal{V}_{\Lambda}(\mathbf{l})$ (et aussi de simplifier le calcul de l’espérance de distorsion).

Le codage est alors réalisé comme suit. Le vecteur $\alpha\mathbf{x} + \mathbf{u}$ est quantifié avec la lattice fine Λ : $\mathbf{x}_q = Q_{\Lambda}(\alpha\mathbf{x} + \mathbf{u})$, $\alpha = \sqrt{1 - D/\sigma_Z^2}$ étant le coefficient d’estimation linéaire optimale de X au décodeur. Finalement, l’indice de $\mathbf{v} = \mathbf{x}_q \bmod \Lambda' = \mathbf{x}_q - Q_{\Lambda'}(\mathbf{x}_q)$ est transmis au décodeur, ce qui nécessite $\log \frac{V'}{V} \approx \frac{n}{2} \log \frac{\sigma_Z^2}{D}$ bits, où V' (respectivement V) est le volume d’une région de Voronoï de lattice Λ' (respectivement Λ) :

$$V' = \int_{\mathcal{V}_{\Lambda'}(0)} dx. \quad (2.19)$$

Le décodeur commence par décoder \mathbf{v} . Ensuite, l’estimation de \mathbf{x} sachant le vecteur de l’information adjacente \mathbf{y} , est définie par :

$$\hat{\mathbf{x}} = \mathbf{y} + \alpha\hat{\mathbf{w}}, \quad (2.20)$$

$$\text{où } \hat{\mathbf{w}} = (\mathbf{v} - \mathbf{u} - \alpha\mathbf{y}) - Q_{\Lambda'}(\mathbf{v} - \mathbf{u} - \alpha\mathbf{y}). \quad (2.21)$$

Le schéma de codage de Zamir *et al.* est présenté sur la figure 2.12. Il est montré dans [ZSE02] que ce schéma approche la fonction débit-distorsion de Wyner-Ziv si le nombre de dimensions n tend vers l’infini.

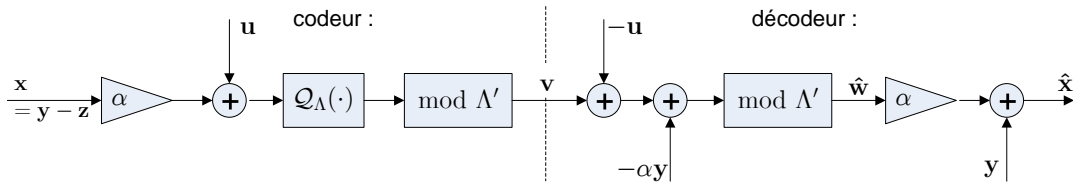


FIG. 2.12 – Schéma de codage de Zamir *et al.*

Servetto [Ser00, Ser07] considère quasiment le même schéma de décodage, avec deux lattices emboîtées Λ et Λ' . Cependant, contrairement à l’approche de Zamir *et al.*, ici il est proposé de trouver un facteur s pour mettre à l’échelle les lattices Λ et Λ' (pour obtenir $s\Lambda$ et $s\Lambda'$ respectivement) au lieu d’appliquer le coefficient α au vecteur \mathbf{x} . De plus, le vecteur de “dither” n’est pas utilisé, ce qui rend le calcul de distorsion plus compliqué. En revanche, les résultats obtenus par Servetto sont plus généraux que ceux de Zamir *et al.*

Xiong *et al.* dans [XLCL03, LCLX04] évaluent les performances de l'approche de Zamir *et al.* utilisée dans un codeur réel. D'abord ils remarquent qu'il existe une corrélation entre l'indice \mathbf{u} et l'information adjacente \mathbf{y} (surtout quand le nombre de dimensions du quantificateur en lattice est faible), et proposent de coder \mathbf{u} avec un codeur de Slepian-Wolf pour atteindre le débit $H(\mathbf{U}|\mathbf{Y}) \leq H(\mathbf{U}) = \log \frac{V'}{V}$.

Ils montrent qu'avec un codeur de Slepian-Wolf "idéal" (atteignant le débit théorique $H(\mathbf{U}|\mathbf{Y})$), la quantification en lattice emboîtée à une dimension (ou deux) est dans l'écart de 1.53 dB (respectivement 1.36 dB) de la fonction débit-distorsion $D_{WZ}^*(R)$ de Wyner-Ziv. En même temps, la performance d'un schéma purement pratique utilisant les codes LDPC pour effectuer le codage de Slepian-Wolf de \mathbf{u} ne montre presque aucun avantage dans le cas de 2 dimensions, même pour les débits élevés (4-5 bits par symbole). La performance de ce schéma pratique est éloigné de 2-3 dB de la fonction débit-distorsion $D_{WZ}^*(R)$.

2.2.2 Techniques de codage DISCUS

Dans [PR03a] un schéma de codage de Wyner-Ziv est proposé (voir la figure 2.13). Il consiste en 5 fonctions M_1, M_2, M_3, M_4 et M_5 . Ces fonctions jouent les rôles suivants :

- La fonction M_1 définit la répartition de l'espace \mathbb{R}^L en 2^{LR_s} intervalles non superposés (intervalles de quantification) :

$$M_1 : \mathbb{R}^L \longrightarrow \{1, 2, \dots, 2^{LR_s}\}, \quad (2.22)$$

où L est la longueur de la séquence X , et R_s est le *débit de source*.

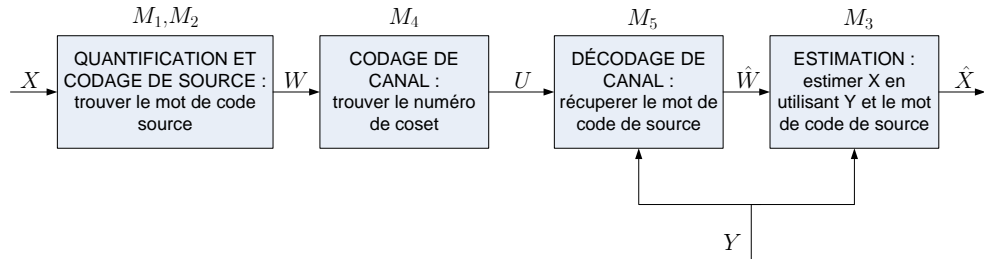


FIG. 2.13 – Schéma de codage de Wyner-Ziv DISCUS

- Pour chaque intervalle de quantification $\{\Gamma_1, \Gamma_2, \dots, \Gamma_{2^{LR_s}}\}$ la fonction M_2 met en correspondance un mot de code (un vecteur dans \mathbb{R}^L), appelé mot de code de source :

$$M_2 : \{1, 2, \dots, 2^{LR_s}\} \longrightarrow \mathbb{R}^L. \quad (2.23)$$

- La fonction $M_3 : \mathbb{R}^L \times \{1, 2, \dots, 2^{LR_s}\} \longrightarrow \mathbb{R}^L$ représente la meilleure estimation de X par décodeur connaissant son intervalle de quantification Γ_i et l'information adjacente Y :

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{a} \in \mathbb{R}^L} \mathbb{E} [d(\mathbf{x}, \mathbf{a}) | \mathbf{x} \in \Gamma_i, \mathbf{Y} = \mathbf{y}], \quad (2.24)$$

où $d(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{L} \sum_{i=1}^L (x_i - \hat{x}_i)^2$ est la fonction de distorsion quadratique.

– La paire M_4 et M_5 définit le codage et décodage de canal :

$$M_4 : \{1, 2, \dots, 2^{LR_s}\} \longrightarrow \{1, 2, \dots, 2^{LR}\}, \quad (2.25)$$

$$M_5 : \mathbb{R}^L \times \{1, 2, \dots, 2^{LR}\} \longrightarrow \{1, 2, \dots, 2^{LR_s}\}, \quad (2.26)$$

avec $R = R_s - R_c$, où R_c est appelé *le débit de canal*. En effet, après la quantification par M_1 , entre le mot de code W généré par M_2 et l'information adjacente Y il y a toujours de la corrélation. L'entropie conditionnelle de W sachant Y est définie par :

$$H(W|Y) = H(W) - I(W; Y), \quad (2.27)$$

où $I(W; Y)$ est l'information mutuelle entre W et Y . Cette information mutuelle peut être vue comme une réduction de débit de source $R_s = H(W)$, procurée par le canal virtuel de corrélation entre W et Y (dont la capacité $R_c = I(W; Y)$).

Ainsi, le processus de construction d'un "bon" code de Wyner-Ziv selon le schéma DISCUS peut se résumer en ces règles de construction :

1. M_1 et M_3 : le but est de minimiser le débit de source R_s sous contrainte que la distorsion reste en-dessous d'un seuil prédéfini D ;
2. M_2 : le but est de maximiser $I(W; Y)$. Comme approximation, pour l'intervalle Γ_i on peut choisir le vecteur \mathbf{w} dont la région de Voronoï est Γ_i ;
3. M_4 : maximiser le débit R_c de code de canal sous contrainte que la probabilité d'erreur reste au-dessous d'un seuil prédéfini ;
4. M_5 : minimiser la complexité de calcul.

Précisons quelques solutions pratiques proposées dans [PR03a] pour le choix de M_1 - M_5 . Pour la quantification (M_1 et M_3), les auteurs considèrent deux possibilités : quantificateur scalaire uniforme ou quantificateur codé en treillis (Trellis Coded Quantizer, TCQ [MF90]), la dernière approche étant plus efficace car la corrélation entre les symboles est exploitée pour atteindre le débit R_s plus bas.

Pour la construction des cosets (le codage de canal, M_4 et M_5), deux possibilités sont également proposées : construction des cosets sans mémoire (comme dans le cas de codage de Slepian-Wolf, voir la section 2.1.2.1), et construction des cosets basée sur les codes en treillis. L'idée de la dernière approche est de faire le codage de canal d'une manière plus efficace, en exploitant la corrélation entre les mots de code de source, générés par M_2 .

Parmi les 4 combinaisons possibles (quantification scalaire ou TCQ, cosets sans mémoire ou TCQ) le meilleur résultat est atteint par la configuration utilisant les codes en treillis dans le deux cas. Il est supérieur de 2-3 dB à celui du codeur n'utilisant les codes en treillis que pour la construction des cosets, avec la quantification scalaire. La distance entre ce résultat et la fonction débit-distorsion est comprise entre 2 dB (à bas débit de 1 bit/symbole) et 3.17 dB (à débit moyen de 2 bits/symbole).



Dans cette section nous présentons le dernier type (selon notre classification) de codage de Wyner-Ziv dont la caractéristique principale est la séparation complète des deux étapes suivantes : la quantification et le codage de Slepian-Wolf. Nous décrivons le premier schéma de ce type, à savoir celui proposé par Aaron et Girod en 2002, ainsi que quelques améliorations introduites plus récemment.

Cette approche est proposée par Aaron et Girod dans [AG02]. Dans le même article ils ont traité les deux problèmes - du codage Slepian-Wolf et du codage Wyner-Ziv. Ici, dans le cas du codage Wyner-Ziv ils suivent le même chemin général que dans leur approche de codage de Slepian-Wolf présenté dans la section 2.1.2.4. Notamment, ils laissent le codeur turbo exploiter la corrélation entre la source (la version quantifiée X_Q de X dans le cas Wyner-Ziv) et l'information adjacente (Y) sans créer les cosets d'une manière explicite.

Dans un premier temps, les valeurs continues de X sont quantifiées avec un quantificateur scalaire uniforme à 2^M niveaux pour obtenir une séquence de valeurs discrètes X_Q prenant les valeurs dans l'ensemble $\{0, 1, \dots, 2^{M-1}\}$. Soit L la longueur de la séquence X_Q à coder (ou longueur sélectionnée d'un bloc s'il s'agit d'un codage par blocs). Alors,

il suffit de 2^{ML} bits pour représenter la séquence X_Q comme une séquence binaire X_B . Cette séquence X_B de longueur 2^{ML} est ensuite codée avec le codeur de Slepian-Wolf (voir la section 2.1.2.4), à savoir les deux séquences de bits de parité X_p^1 et X_p^2 sont générées avec deux codeurs récurrents convolutionnels (CRS) identiques, la séquence d'entrée X'_B du deuxième CRS étant une version entrelacée de X_Q , voir le schéma à la figure 2.14.

Au niveau du décodage, les probabilités de canal des *symboles* (au lieu des *bits*) sont calculées d'abord en utilisant l'information adjacente Y (calculée via la fonction $f(y|x)$) et deux séquences de bits de parité X_p^1 et X_p^2 . Ensuite, le décodage itératif se fait avec les deux décodeurs SISO (Soft-In-Soft-Out) utilisant l'algorithme BCJR [BCJR74], qui s'échangent une information extrinsèque sur des symboles X_Q . Après quelques itérations, l'estimation finale \hat{X}_Q est définie par la probabilité *a posteriori* de la dernière itération.

L'analyse des performances de ce schéma [AG02] montre une amélioration considérable par rapport au premier schéma DISCUS proposé dans [PR99]. Cependant, par rapport au schéma DISCUS amélioré proposé dans [PR03a] (utilisant la construction des cosets et la quantification avec des codes en treillis), l'avantage de l'approche de Aaron et Girod utilisant les codes turbo, n'est pas très important.

Ce schéma fut amélioré dans les travaux plus tardifs de Girod *et al.* Par exemple, dans [ARSG04] ils proposèrent la reconstruction optimale de X en termes de l'erreur quadratique moyenne sachant l'indice de quantification Q et l'information adjacente Y , qui pour une fonction de distorsion quadratique $d(x, \hat{x}) = (x - \hat{x})^2$ est définie par :

$$\hat{x}_{\text{opt}} = \mathbb{E}_{f(x|y)} [x|Q = q, Y = y]. \quad (2.28)$$

De plus, les auteurs proposent de coder la séquence X_Q par M plans de bits de longueur L , et non pas en une seule séquence binaire X_B de longueur ML , en modélisant la corrélation entre les plans de bits X_Q^l et Y_Q^l , ($l = 1, \dots, M$) comme un canal binaire symétrique, ce qui permet de mieux exploiter les dépendances entre les plans de bits. En effet, dans ce cas les plans de bits sont décodés successivement, et chaque décodage peut profiter de la connaissance des plans de bits précédents déjà décodés.

En 2005, Varodayan *et al.* [VAG05] proposent d'utiliser les codes LDPC au lieu des codes turbo dans le même schéma pour mieux s'approcher de la borne de Slepian-Wolf.

2.2.3.2 Quantification optimale

Dans un autre article, Rebollo-Monedero *et al.* [RMZG03] s'intéressent au problème de construction d'un quantificateur optimal en terme de débit-distorsion pour un schéma de codage de Wyner-Ziv, i.e. un quantificateur qui minimise le coût de codage $(1 - \lambda)D + \lambda R$ avec la distorsion $D(X, \hat{X}) = \frac{1}{L} \sum_{i=1}^L d(x_i, \hat{x}_i)$ et le débit R de codage de Slepian-Wolf (dans le cas idéal $R = H(Q|Y)$, où Q est l'indice de quantification).

Pour cela, ils étendent l'algorithme itératif de Lloyd [Llo82], souvent utilisé dans la construction des quantificateurs optimaux dans le cas non-distribué.

Contrairement au cas non-distribué, dans le cas Wyner-Ziv cet algorithme peut converger vers un quantificateur optimal dont les intervalles sont disjoints. Cela s'explique par la présence de l'information adjacente Y au décodeur. En effet, si la distance entre deux intervalles ayant le même l'indice de quantification q , est beaucoup plus importante que l'espérance de $|X - Y|$, le décodeur peut choisir le bon intervalle avec une probabilité d'erreur assez faible. En même temps, l'affectation d'un même indice q à plusieurs intervalles permet d'utiliser des intervalles plus courts (ce qui signifie une distorsion D plus basse) sans augmenter l'entropie de Q $H(Q)$.

L'algorithme de Lloyd généralisé se résume alors aux étapes suivantes :

1. choisir un schéma de quantification de départ $Q^{(1)}$ avec les fonctions de quantification $q^{(1)}(x)$; fixer $k = 1$;
2. trouver les fonctions de reconstruction $\hat{x}^{(k)}(q, y)$ optimales pour $q^{(k)}(x)$;
3. calculer le débit $R^{(k)}$ (par exemple $R^{(k)} = H(Q^{(k)}|Y)$) et la distorsion $D^{(k)}$ pour les fonctions de quantification $q^{(k)}(x)$ et les fonctions de reconstruction $\hat{x}^{(k)}(q, y)$; calculer le coût de codage $J^{(k)} = (1 - \lambda)D^{(k)} + \lambda R^{(k)}$;
4. si $J^{(k)} > J^{(k-1)}$ l'algorithme s'arrête, sinon trouver le nouveau schéma $Q^{(k+1)}$ avec les fonctions de quantification $q^{(k+1)}(x)$ optimales pour les fonctions de reconstruction $\hat{x}^{(k)}(q, y)$, et revenir à l'étape 2 avec $k = k + 1$.

La convergence de l'algorithme vers un point fixe est assurée. Mais ce point fixe ne correspond pas nécessairement à un minimum global.

Les résultats pour les sources Gaussiennes dans [RMZG03] montrent que la distorsion avec un quantificateur construit par l'algorithme proposé (quantificateur distribué) est plus basse qu'avec d'autres quantificateurs construits sans prendre en compte la corrélation entre X et Y (quantificateurs non-distribués), mais utilisant le même nombre d'indices de quantification (comme expliqué ci-dessus, le nombre d'intervalles peut être différent du nombre d'indices pour le quantificateur distribué). Il s'agit alors de l'optimisation de la distorsion seule, c'est-à-dire avec $\lambda = 0$.

Cependant, dans le cas où $\lambda > 0$ où le débit est donné par le théorème de Slepian-Wolf ($R = H(Q|Y)$), cette optimisation n'a que peu d'impact sur les performances finales, c'est-à-dire les quantificateurs non-distribués et uniformes donnent quasiment le mêmes résultats en terme de débit-distorsion. Cela s'explique par le fait qu'une quantification plus fine de X qui réduit la distorsion D , réduit également la corrélation entre Q et Y , et comme le codage exploite idéalement cette corrélation, le débit $R = H(Q|Y)$ se trouve augmenté aussi.

2.3 Codage distribué de vidéo

Les schémas de codage de Slepian-Wolf et Wyner-Ziv présentés dans les deux chapitres précédents n'avaient pour but que de confirmer la possibilité de construction pratique d'un système de codage de sources distribuées. Bien qu'il y ait actuellement des pertes en performance de ces systèmes par rapport aux systèmes de codage conventionnels (contrairement aux théorèmes de Slepian-Wolf et Wyner-Ziv), cette différence est en diminution constante. En contrepartie d'une légère sous-optimalité de compression, les systèmes de codage de sources distribuées ont un avantage considérable par rapport aux systèmes traditionnels : l'absence de communication entre les codeurs, et par conséquent la complexité faible de codage.

Une application originale des principes de codage distribué est l'application à la compression de vidéo. Les premiers systèmes ont été proposés en 2002, peu après les premiers schémas concrets de codage de Wyner-Ziv. Dans ces systèmes, les images successives de la vidéo sont considérées comme les sources corrélées, et par conséquent codées séparément. Cette idée qu'on peut faire l'estimation de mouvement au décodeur, mais non pas au codeur, avait séduit beaucoup de chercheurs. En résumé, le codage distribué de vidéo peut proposer les avantages suivants par rapport aux codeurs traditionnels de vidéo :

- répartition flexible de la complexité entre le codeur et le décodeur ;
- résistance naturelle aux erreurs de transmission ;
- extensibilité du système (car les niveaux suivants ne dépendent pas du décodage précis des niveaux précédents) ;
- orientation naturelle vers la compression de multivue, c'est-à-dire des images ou des vidéos de la même scène filmées de points de vue différents.

Ces avantages sont appréciés dans plusieurs domaines d'application, notamment les domaines où les appareils qui doivent effectuer le codage de vidéo, ont des contraintes sur la complexité, la puissance ou l'énergie disponibles (téléphones portables, caméscopes de surveillance vidéo sans fil etc.). En revanche, il est supposé que le décodeur a suffisamment de puissance pour faire le décodage conjoint, qui est (dans les systèmes actuels) beaucoup plus complexe que le codage conjoint dans les systèmes de codage vidéo traditionnels.

Nous commencerons cette section par la présentation de deux architectures qui furent les premiers systèmes proposés de codage vidéo, à savoir PRISM [PR02] et l'architecture de Stanford [AZG02]. Puis nous présenterons des contributions qui ont été proposées depuis le début des travaux dans le domaine de codage distribué de vidéo. Ensuite, nous présenterons en détails le codeur DISCOVER, qui à ce jour est une des mises en œuvre les plus développées et efficaces des principes de codage distribué de vidéo [dis07]. L'intérêt du codeur DISCOVER est aussi suscité par le fait que ce système a servi de plate-forme pour l'évaluation de l'efficacité des contributions de cette thèse, dont le résumé sera présenté ensuite.

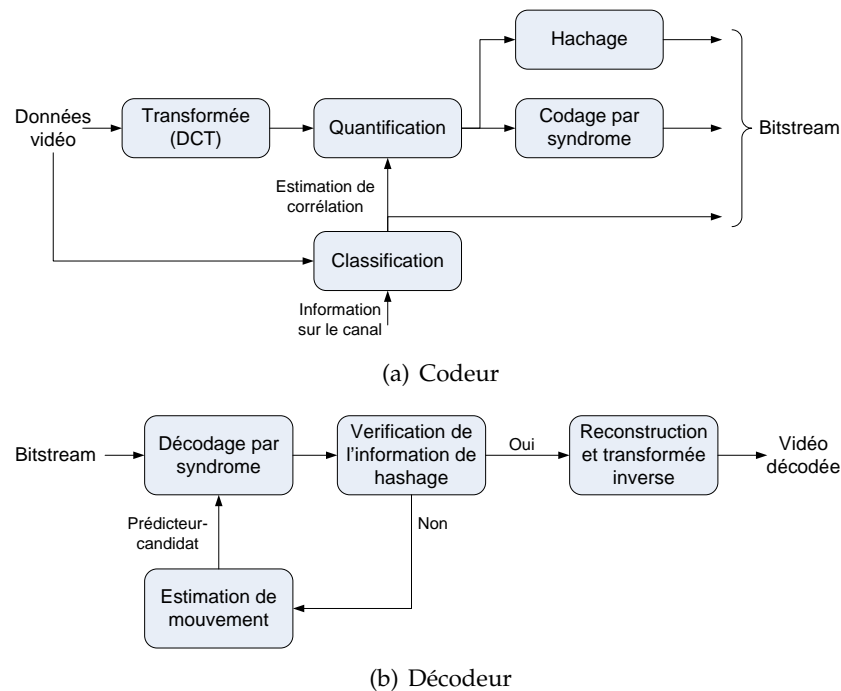


FIG. 2.15 – Architecture de PRISM

2.3.1 Système de codage vidéo distribué PRISM

L'architecture surnommée PRISM [PR02, PMR07] (de l'anglais "Power-efficient, Robust, hIgh compression, Syndrome-based Multimedia coding") fut proposée par un groupe de chercheurs de l'université de Berkeley en 2002. C'est une approche basée sur la représentation par blocs. Ainsi, chaque image vidéo est codée par blocs de taille 8x8 ou 16x16. Ces blocs peuvent avoir des modes de codage différents, par exemple : le mode "skip" (aucun codage, copier le bloc de l'image de référence), le mode "intra" (codage traditionnel exploitant la corrélation spatiale), le mode "Wyner-Ziv" (codage de Wyner-Ziv basé syndromes). Cette variété de modes permet de choisir le meilleur mode de codage localement étant donné la corrélation dans les domaines temporel et spatial, tout en essayant de garder la complexité du codeur à un niveau bas, c'est-à-dire en donnant la préférence au mode "Wyner-Ziv". Ci-dessous nous expliquons le processus de codage/décodage avec PRISM pas-à-pas comme il est présenté dans [PMR07] (voir la figure 2.15).

2.3.1.1 Processus de codage

Le schéma du codeur PRISM est présenté sur la figure 2.15(a). Le processus de codage comporte les étapes suivantes :

1. **Classification** - C'est l'étape la plus importante du processus de codage. Le

choix de classe dépend de la corrélation (l'erreur moyenne quadratique E du résidu) entre le bloc en question et le bloc de référence. Suivant la complexité maximale autorisée au codeur, le bloc de référence (appelé prédicteur) peut être le bloc co-positionné dans l'image précédente (complexité minimale), ou un bloc trouvé avec une technique d'estimation de mouvement simple (complexité modérée) ou avancée (complexité du codeur est proche de celle des codeurs conventionnels prédictifs). Dans les deux derniers cas, le vecteur de mouvement peut être communiqué au décodeur.

Si le meilleur prédicteur est disponible au codeur (trouvé avec une technique d'estimation de mouvement avancée), la corrélation entre le bloc actuel X et le prédicteur (l'information adjacente) Y peut être déterminée d'une manière précise, ainsi que le nombre de plans de bits à coder par syndromes (c'est-à-dire en mode "wyner-ziv", voir ci-dessous).

Si le codeur a une contrainte de complexité, le prédicteur n'est qu'une estimation du meilleur prédicteur qui sera disponible au décodeur, l'une des classes prédéfinies N^i est choisie par seuillage de E (16 classes au total sont proposées dans [PMR07]). Chaque classe N^i modélise le bruit de corrélation $Z = X - Y$ comme un ensemble de variables aléatoires Laplaciennes $\{N_0^i, N_1^i, \dots, N_{63}^i\}$ d'espérance 0 et de variance $1/\alpha_f^2$, $f = 0, \dots, 63$, qui représentent chacune la corrélation entre les coefficients respectifs de transformée des blocs (voir ci-dessous point **Transformée**). Les classes et leurs paramètres, ainsi que les seuils, sont déterminés préalablement pendant une étape d'entraînement d'algorithme.

Dans tous les cas, le but de cette étape est de définir combien de bits de chaque coefficient du bloc seront (par ordre croissant de niveau de corrélation) :

- codés avec le codage entropique sans utiliser le prédicteur Y (les bits les moins importants, LSB en anglais) ;
- codés avec le code de canal basé sur les cosets (le codage de Slepian-Wolf) ;
- rejetés et non codés (pour utiliser les bits respectifs du prédicteur Y pendant la reconstruction).

Pour un exemple de bloc ces trois groupes ci-dessus sont représentés schématiquement sur la figure 2.16 par les couleurs noire, grise et blanche (dans le même ordre).

2. **Transformée** - Le but de cette étape est de réduire la corrélation spatiale à l'intérieur du bloc. La transformée en cosinus discrète (en anglais "Discrete Cosine Transform", DCT) des blocs de taille 8x8 est utilisée comme une approximation de la transformée de Karhunen-Loève (KLT). Les coefficients sont scannés en zigzag (dans le domaine 2D des coordonnées du bloc) pour former un vecteur unidimensionnel constitué de 64 coefficients de fréquence, arrangés dans l'ordre décroissant de l'énergie.
3. **Quantification** - Un quantificateur scalaire est ensuite appliqué au vecteur de coefficients. Les pas de quantification sont fixés de telle manière que la distorsion

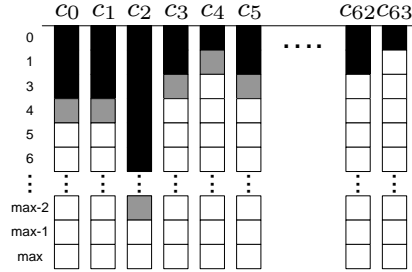


FIG. 2.16 – Représentation des plans de bits pour un vecteur de 64 coefficients. Les plans de bits sont arrangés dans l'ordre croissant de l'importance : le plan de bits 0 groupe les bits les moins importants, le plan de bits max - les bits les plus importants des coefficients c_0, c_1, \dots, c_{63} .

attendue soit acceptable. Notons que la stratégie peut varier selon la classe du bloc pour prendre en compte la corrélation entre bloc X et le prédicteur Y .

4. **Codage par syndrome** - Un bloc comprend les plans de bits codés entropiquement plus le syndrome des plans de bits codés avec un code de canal (la répartition des plans de bits est définie pendant la classification).
 - *Le codage entropique* se fait sur un alphabet fabriqué des 4-uplets (**run**, **depth**, **path**, **last**) [CEGK98] pour chaque coefficient DCT, où **run** signifie le nombre de coefficients pour lesquels le nombre de plans de bits codés est nul (parmi les coefficients précédents dans ce même bloc); **depth** signifie le nombre de plans de bits codés entropiquement pour le coefficient actuel; **path** représente ces bits (avec $\text{path} \in \{0, \dots, 2^{\text{depth}} - 1\}$); et **last** est 1 pour le dernier coefficient dans le bloc dont le nombre de plans de bits codés est non-nul, et 0 sinon. Pour coder la séquence de symboles ainsi créée, le codeur arithmétique (voir la section 1.2.1.2) est utilisé dans [PMR07].
 - *Le codage de canal basé sur les cosets*. Le reste des bits choisis pour le codage (à l'exception de ceux codés entropiquement) est codé avec un code de canal basé sur les cosets (voir section 2.1.2.1). A chaque classe N^i correspond la matrice de parité H_i d'un code correcteur d'erreurs linéaire (n, k_i) . Le syndrome s est défini comme $s = H_i b$, où b est une séquence de bits d'entrée. Le taux de code est par conséquent $\frac{n-k_i}{n}$ bits par coefficient, et est la fonction du niveau estimé de corrélation. Les codes de famille Bose, Ray-Chaudhuri et Hocquenghem (BCH) [MS77] sont utilisés dû à leur efficacité pour des séquences courtes.
5. **Hachage** - Pour aider le décodeur à trouver le bon prédicteur (comme expliqué ci-dessous dans la section "Décodage"), une information de hachage est envoyée pour chaque bloc. Cette information sert de signature et permet de déterminer quand le décodage est réussi. Dans [PMR07] une somme CRC (Cyclic Redun-

dancy Check) [Wil] de longueur 16 bits calculée sur les coefficients quantifiés, est proposée comme information de hachage.

2.3.1.2 Processus de décodage

Le décodeur PRISM est présenté sur la figure 2.15(b). Le processus de décodage comporte les étapes suivantes :

1. **Estimation de mouvement** - Le but de cette étape est de générer le meilleur prédicteur Y pour le bloc actuel à décoder. En réalité, le module génère plusieurs prédicteurs, qui sont essayés les uns après les autres pour le décodage par syndrome (voir ci-dessous) jusqu'à ce que l'information de hachage du bloc décodé corresponde à celle reçue. Dans [PMR07], l'ensemble des prédicteurs-candidats est formé par tous les blocs d'une fenêtre de recherche centrée sur le bloc co-positionné dans l'image précédente, avec une précision d'un demi-pixel.
2. **Décodage par syndrome** - Cette étape comporte deux sous-étapes. Dans un premier temps le coset du bloc est trouvé. Le coset est défini par le syndrome du bloc qui comprend les plans de bits codés entropiquement (couleur noire dans la figure 2.16), et, s'il y a des plans de bits codés avec un code de canal (couleur grise dans la figure 2.16), par le syndrome de ceux-ci également. Dans un deuxième temps, dans ce coset le mot de code (qui comprend l'ensemble des coefficients quantifiés) le plus proche du prédicteur Y est choisi (des algorithmes sous-optimaux peuvent être utilisés pour réduire la surcharge totale de décodage).
3. **Vérification de l'information de hachage** - Après le décodage par syndrome, l'information de hachage est calculée sur le bloc décodé. Si elle ne correspond pas à celle reçue du codeur, l'algorithme revient à l'étape d'estimation de mouvement, et le prédicteur suivant est sélectionné.
4. **Reconstruction et transformée inverse** - Après avoir décodé la version quantifiée du bloc avec une probabilité faible d'erreur ce qui est garanti par l'information de hachage, le décodeur l'utilise ensemble avec le meilleur prédicteur Y^* (en termes de l'erreur quadratique moyenne) pour estimer la version originale non-quantifiée du bloc. La statistique de distribution des coefficients (si elle est disponible) peut être également exploitée. Ensuite, la transformée DCT inverse est appliquée à cette estimation pour obtenir le bloc dans le domaine pixel, constituant une portion de l'image décodée.

2.3.2 Système de codage vidéo distribué de Stanford

Une autre architecture de codage vidéo distribué [AZG02] fut développée dans l'université de Stanford au même moment que celle de PRISM. Contrairement à cette

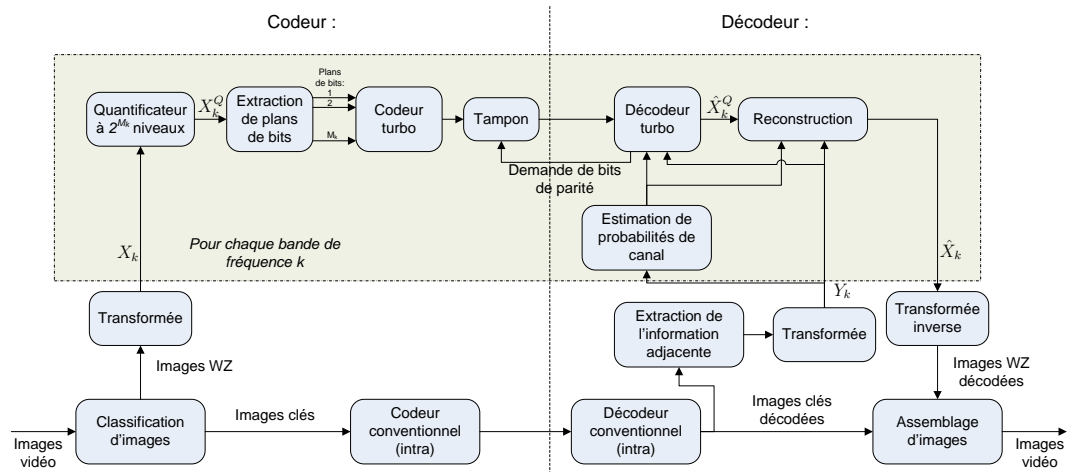


FIG. 2.17 – Architecture de Stanford (domaine transformé)

dernière, l'architecture de Stanford est orientée vers le codage par images et non par blocs. La première version [AZG02] supposait le codage dans le domaine pixel, mais peu de temps après, des solutions dans le domaine transformé ont été également proposées [ARSG04, GARRM05]. L'avantage de ces méthodes est une compression plus efficace grâce à l'exploitation de la corrélation spatiale.

Nous ne présentons ici que l'approche dans le domaine transformé qui, à présent, a quasiment supplanté l'approche pixel dans la littérature. Cependant, le lecteur s'y intéressant peut trouver les détails dans [AZG02, ASG03, GARRM05].

Le schéma du système de codage de [ARSG04] est présenté sur la figure 2.17. On remarque tout de suite une différence importante avec celui de PRISM : la présence de la voie de retour. Cette solution permet de mieux contrôler le débit (car c'est le décodeur qui le fait), et par conséquent d'avoir la meilleure performance finale en terme de débit-distorsion. Cependant le domaine d'utilisation du système se trouve restreint aux applications qui nécessitent un décodage en temps réel, et par ailleurs la complexité du décodeur augmente car il doit alors demander les bits de parité par des petites portions pour un contrôle de débit plus efficace.

Ci-dessous nous détaillons les processus de codage et de décodage dans l'architecture de Stanford.

2.3.2.1 Processus de codage

1. **Classification d'images** - Dans un premier temps, les images de la vidéo sont réparties en groupes (GOP de l'anglais "Group Of Pictures") d'une taille fixe. Chaque GOP comprend une image clé (KF de "Key Frame") suivie d'une ou plusieurs images Wyner-Ziv (WZ). Les images clés sont codées avec un codeur conventionnel n'exploitant que la corrélation spatiale (et non pas temporelle),

par exemple avec des codeurs de standards H.263+ ou H.264/AVC dans le mode "intra", c'est-à-dire sans faire l'estimation de mouvement. Le reste du GOP est codé avec un codeur Wyner-Ziv (voir ci-après).

2. **Transformée** - A l'instar de l'architecture PRISM, chaque image WZ du GOP subit d'abord une transformée DCT sur chaque bloc de taille 4x4 pixels. Ensuite, les coefficients de même fréquence sont groupés ensemble, formant 16 bandes de fréquence de taille $N/16$ chacune (où $N = l \times h$ est la taille de l'image en pixels), les coefficients de chaque bloc étant parcourus en zigzag.
3. **Quantification** - Chaque bande de fréquence est quantifiée avec un quantificateur scalaire uniforme à 2^{M_k} niveaux, $k = 1, \dots, 16$, où les valeurs M_k sont choisies en fonction de la distorsion voulue.
4. **Extraction des plans de bits** - Ensuite, les coefficients quantifiés de chaque bande sont alignés les uns à côté des autres, et M_k plans de bits sont formés par les bits se trouvant au même niveau. Ainsi chaque plan de bits a la taille de la bande entière $N/16$.
5. **Codeur turbo** - A l'étape suivante, chaque plan de bits est codée avec un codeur turbo comme dans le schéma de codage Slepian-Wolf proposé par Aaron et Girod (voir la section 2.1.2.4). Plus précisément, les bits systématiques sont rejetés, tandis que les bits de parité sont mis dans un tampon accessible aux requêtes de décodeur.

2.3.2.2 Processus de décodage

1. **Décodage conventionnel** - Le décodeur commence par décoder les images clés. Comme elles sont codées en mode "intra", elles peuvent être décodées une par une et séparément des autres images (WZ).
2. **Extraction de l'information adjacente** - A partir des images clés des GOPs adjacents et/ou les images WZ déjà décodées, le décodeur construit l'information adjacente par interpolation ou extrapolation de ces images connues, souvent par des méthodes d'estimation de mouvement. Cette étape est décrite avec plus de détails pour le codeur DISCOVER dans la section 2.3.4, ainsi que dans le chapitre 3.

De la même façon que les images WZ, l'information adjacente subit la transformée DCT par blocs 4x4, et 16 bandes de fréquence sont formées.

3. **Estimation des probabilités de canal** - A l'instar du schéma PRISM, l'erreur entre l'information adjacente Y_k et l'image codée X_k est modélisée (par bandes de fréquence, $k = 1, \dots, 16$) comme un canal virtuel avec un bruit additif Laplacien : $Y_k = X_k + Z_k$, où $p_{Z_k}(x) = \frac{\alpha_k}{2} e^{-\alpha_k |x|}$. Par conséquent, les probabilités de canal $p_{k,i}^b$

pour chaque plan de bits b ($1 \leq b \leq M_k$) sont calculées comme suit :

$$p_{k,i}^b = \Pr(x_{k,i}^b = 1 | y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1}) = \frac{\int_{Q(1)} \frac{\alpha_k}{2} e^{-\alpha_k |x|} dx}{\int_{Q(0) \cup Q(1)} \frac{\alpha_k}{2} e^{-\alpha_k |x|} dx}, \quad (2.29)$$

où $x_{k,i}^b$ représente le bit à la position i dans le plan de bits b de la bande de fréquence k , $y_{k,i}$ représente le coefficient DCT de Y à la position i dans la bande de fréquence k , et $Q(m) = Q(m, x_{k,i}^1, \dots, x_{k,i}^{b-1})$ est l'ensemble de tous les intervalles de quantification correspondants aux symboles x^q quantifiés dont les b bits les plus importants sont $(x_{k,i}^1, \dots, x_{k,i}^{b-1}, m)$ (avec une incertitude sur les $(M_k - b)$ bits les moins importants).

Dans les travaux [ARSG04, GARRM05] les paramètres α_k du modèle de canal sont supposés connus au décodeur (estimés préalablement hors-ligne).

4. **Décodage turbo** - Étant données les probabilités de canal et les bits de parité reçus du codeur, le décodage turbo se fait par plans de bits, en commençant par le plan de bits $b = 1$ le plus important ("Most Significant Bit", MSB). Au début, le décodeur n'a qu'une petite portion de bits de parité, le reste étant poinçonné. Une itération complète de décodage turbo est faite, dont le résultat est comparé au plan de bits original pour déterminer le taux d'erreurs (en pratique cette comparaison n'est pas réalisable, et dans le schéma DISCOVER une autre méthode d'estimation du taux d'erreurs a été proposée, voir la section 2.3.4). Si le taux d'erreurs est supérieur à un seuil prédéfini (e.g. 10^{-3}), une autre portion de bits de parité est transférée du tampon au décodeur réduisant ainsi le taux de poinçonnage des codeurs CRS constituant le codeurs turbo, et le décodage turbo est relancé.

Notons que les portions de bits de parité doivent être suffisamment petites pour que le contrôle de débit soit efficace. Cela implique un nombre important de cycles de décodage turbo, et par conséquent une complexité élevée. Nous abordons ce problème dans une contribution (voir section 4.4).

5. **Reconstruction** - Après avoir décodé tous les M_k plans de bits d'une bande k , le décodeur passe à la reconstruction de la bande k . Pour cela, les plans de bits sont regroupés pour estimer les symboles quantifiés $x_{k,i}^q$. La meilleure estimation en termes d'erreur quadratique moyenne est donnée par :

$$\hat{x}_{k,i} = \mathbb{E} \left[x \mid x \in Q(x_{k,i}^q), y_{k,i} \right], \quad (2.30)$$

où $Q(x_{k,i}^q)$ est l'intervalle de quantification correspondant à $x_{k,i}^q$, et l'espérance est calculée sur la loi de probabilité jointe $p_{xy}(x, y_{k,i})$.

L'obtention des formules directes pour une reconstruction optimale en terme de l'erreur quadratique moyenne fait l'objet d'une contribution de cette thèse (voir la section 4.3).

6. **Transformée inverse** - La transformée DCT inverse est réalisée sur les bandes de fréquence \hat{X}_k reconstruites, ainsi que sur les bandes de fréquence de l'information adjacente Y_k copiées à la place des bandes pour lesquels aucun plan de bits n'a été transmis (c'est-à-dire à la place des bandes k_0 telles que $M_{k_0} = 0$).
7. **Assemblage des images** - Finalement, les images décodées (les images clés ainsi que les images WZ) sont assemblées dans le bon ordre pour former la vidéo décodée.

2.3.3 Diverses améliorations du système de codage de Stanford

L'architecture de Stanford a été plus largement reprise et utilisée que celle de Berkeley (PRISM). Ce choix est certainement lié au fait que plus d'information est disponible sur cette architecture (surtout dans [GARRM05]). On peut également souligner sa faible complexité de mise en œuvre par rapport à PRISM.

Par conséquent, de nos jours elle semble être plus complète et plus performante que celle de PRISM [dis07]. Nous présentons dans cette section les travaux complémentaires majeurs qui y ont contribué.

2.3.3.1 Choix adaptatif de la taille de GOP

Au lieu de fixer la taille de GOP à une constante pour toute la séquence comme dans [AZG02, ARSG04], Ascenso *et al.* proposent dans [ABP06] un algorithme de choix adaptatif et local de la taille de GOP en fonction du contenu de la vidéo. Cet algorithme est réalisé comme une étape supplémentaire avant le codage de la séquence, qui définit les tailles de tous les GOP de la séquence. L'idée générale est de commencer par des GOP de taille minimale (de taille 1, c'est-à-dire un GOP sans images WZ), et de joindre ensuite les GOPs adjacents (réalisant ainsi un GOP d'une taille plus grande) tant que la corrélation entre ses images le permet. La corrélation est estimée sans estimation de mouvement complexe, mais par calcul des moments statistiques (e.g. des histogrammes de luminance) de chaque image, ce qui permet en les comparant pour deux images de vidéo de prendre une décision sur l'activité entre ces deux images, et en conséquence d'estimer la corrélation entre eux.

2.3.3.2 Quantification

Des efforts ont été faits pour améliorer le module de quantification. Par exemple, les quantificateurs en lattice emboîtée optimaux ont été proposés dans [FE01, FZE04]. Malheureusement, la complexité des quantificateurs en lattice emboîtée à plusieurs dimensions est trop élevée pour des mises en œuvres effectives, tandis qu'à nombre faible de dimensions ils sont sous-optimaux.

Le problème de construction d'un quantificateur scalaire optimal en terme de débit-distorsion est traité dans [RMZG03] via l'algorithme dont nous avons déjà parlé dans

le section 2.2.3.2. Une variation plus orientée vers l'utilisation pratique de cet algorithme (que l'on peut considérer comme une généralisation de l'algorithme de Lloyd [Llo82]) a été proposée par Sheng *et al.* dans [SXJLW07]. Leur méthode n'optimise la distorsion du quantificateur qu'à débit fixé (en utilisant les images précédentes déjà décodées pour estimer la corrélation). Par conséquent, le taux d'erreur du décodeur turbo est légèrement augmenté, mais la distorsion de la reconstruction est diminuée, et le résultat global semble être plutôt positif selon leurs simulations.

2.3.3.3 Codage de Slepian-Wolf

Le codage de Slepian-Wolf est notamment le cœur d'un système de codage Wyner-Ziv et d'un système de codage vidéo distribué. Par conséquent, le choix de code de canal est très important, et influence non seulement la performance débit-distorsion du système, mais également sa complexité.

A ses débuts le système de Stanford utilisait les codes turbo, mais Varodayan *et al.* [VAG05, VAG06] ont montré que l'utilisation des codes LDPC permet de gagner en performance ainsi qu'en complexité. Dans ces travaux ils également proposent un mécanisme d'adaptation de débit similaire à la technique de poinçonnage utilisée pour les codes turbo, connue sous les nom LDPCA (LDPC Accumulate) et SLDPCA (Sum LDPCA).

La performance d'un code LDPC dépend de sa matrice de parité \mathbf{H} . Aussi, des stratégies avancées de construction des bons codes adaptés au contexte d'utilisation ont récemment été proposées dans [ABP08]. Le lecteur peut revenir aux sections 1.3.3 et 2.1.2.5 pour se rappeler les principes de codage LDPC.

Récemment les codes de fontaine [Mac05] ("Fountain Codes" en anglais) ont commencé à attirer l'attention des chercheurs. Les codes de fontaine ont une propriété que pour toute séquence des symboles d'entrée une autre séquence (de code) des symboles (peut-être de longueur infinie) peut être générée de telle manière que les symboles d'entrée peuvent être récupérés à partir de n'importe quel sous-ensemble de cette séquence de code de longueur légèrement supérieure à la longueur de la séquence d'entrée. C'est pourquoi les codes de fontaine sont aussi appelés les codes d'effacement sans taux (en anglais "Rateless Erasure Codes"). Xu *et al.* [XSX07] considèrent l'utilisation des codes dits "raptor" (le dernier ajout dans la famille des codes de fontaine) dans leur architecture de codage vidéo distribué, conçue pour une transmission extensible de vidéos sur les réseaux sans fil.

2.3.3.4 Amélioration de l'information adjacente

La qualité de l'information adjacente joue aussi un rôle très important dans l'architecture de Stanford. Dans un certain sens, le décodeur PRISM est moins influencé par la méthode de génération de l'information adjacente car il a plusieurs prédicteurs-

candidats parmi lesquels il peut y avoir des prédicteurs suffisamment corrélés avec le bloc codé. Le décodeur de l'architecture de Stanford n'a pas cette possibilité.

Cela a motivé les chercheurs à développer des algorithmes plus complexes et plus efficaces pour extraire l'information adjacente que la première approche qui consistait à faire tous simplement la moyenne des deux images clés les plus proches [AZG02]. Parmi les approches basées sur l'estimation de mouvement par blocs, l'approche d'Ascenso *et al.* [ABP05b, ABP06] est l'une des plus avancées. Elle comporte plusieurs outils comme l'estimation de mouvement exhaustif entre deux images (images clé ou images WZ déjà décodées), le raffinement bidirectionnel des vecteurs de mouvement, le lissage spatio-temporel du champ de mouvement, ou encore l'approche hiérarchique avec deux tailles de blocs différentes (ajouté dans [AP07, AP08]). Faisant partie du système DISCOVER, cette méthode est décrite en détails dans la section 2.3.4.5.

Une analyse plus complète de l'état de l'art se trouve dans l'introduction au chapitre 3 consacré à l'extraction de l'information adjacente.

2.3.3.5 Modélisation de corrélation

Les bonnes performances d'un système de codage de Wyner-Ziv ne sont réalisables qu'avec un bon modèle de corrélation connu au codeur comme au décodeur. En pratique, cela pose un vrai problème aux systèmes réels (qui travaillent avec des sources réelles comme des données vidéo, et non pas des sources théoriques) car le codeur n'a pas accès à l'information adjacente, et le décodeur à son tour n'a pas accès à l'image originale, tandis que pour estimer correctement la corrélation il faut avoir les deux. De plus, pour les données vidéo ce modèle (ses paramètres) change assez souvent selon l'activité et le contenu de la vidéo. C'est pourquoi les premiers systèmes de codage vidéo distribué (notamment [AZG02, GARRM05]) n'ont pas même abordé ce problème supposant (avec peu de réalisme) que le modèle est connu ou estimé préalablement.

Ce problème est traité pour la première fois par Brites *et al.* [BAP06b, BAP06c, BP08], qui ont proposé d'estimer la corrélation au décodeur en analysant l'erreur de compensation de mouvement entre les deux images utilisées pour la construction de l'information adjacente. En supposant la loi de distribution d'erreur Laplacienne, plusieurs niveaux de granularité de modélisation sont prévus, à savoir :

- le niveau image (1 paramètre par image, ou par bande de fréquence dans le domaine transformée) ;
- le niveau bloc (1 paramètre par bloc de pixel, par coefficient de bande de fréquence dans le domaine transformée) ;
- le niveau pixel (n'est utilisé que dans le domaine pixel).

Cette granularité a permis d'améliorer d'une manière non-négligeable la performance du système, car la corrélation en réalité n'est pas stationnaire, et requiert une modéli-

sation locale.

Le problème de modélisation de corrélation fait l'objet du chapitre 4 de cette thèse.

2.3.3.6 Reconstruction

Dans l'architecture de Stanford décrite dans la section 2.3.2.2 nous avons présenté une solution optimale en terme d'erreur quadratique moyenne donnée par (2.30). La solution adoptée dans les systèmes réels est en général plus simple. Elle consiste à prendre comme valeur reconstruite $\hat{x}_{k,i}$ la valeur de $y_{k,i}$ si elle est à l'intérieur de l'intervalle $Q(x_{k,i}^q)$, et sinon la borne de l'intervalle la plus proche de $y_{k,i}$ [AZG02] :

$$\hat{x}_{k,i} = \arg \min_{x \in Q(x_{k,i}^q)} |x - y_{k,i}|. \quad (2.31)$$

La raison d'utiliser cette méthode simple est l'absence de formules directes pour la reconstruction optimale. L'une des contributions présentées dans cette thèse a pour but la dérivation de telles formules pour le modèle Laplacien de corrélation (voir la section 2.4). Cette solution a été adoptée dans le système DISCOVER.

2.3.3.7 Le choix du mode de codage par blocs

L'algorithme de choix adaptatif de la taille de GOP de [ABP06] présentée précédemment, aborde d'une manière efficace le problème de variabilité temporelle du modèle de corrélation. Cependant, sa variabilité spatiale est toujours cause de sous-optimalité de l'architecture de Stanford qui code l'image entière contrairement au codeur PRISM qui prend les décisions au niveau de chaque bloc.

Tagliasacchi *et al.* [TTT⁺06b] proposent une approche (dans le domaine pixel) inspirée en quelque sorte du comportement du codeur PRISM, qui sélectionne le mode "intra" ou le mode "WZ" pour chaque bloc de 8x8 pixels en fonction de la corrélation estimée et de la variance de luminance dans le bloc. Deux solutions sont proposées, concernant notamment la prise de décision au niveau de codeur d'une part et au niveau de décodeur d'autre part. Dans le premier cas la corrélation est estimée en comparant le bloc avec le bloc co-positionné dans l'image précédente, tandis qu'au décodeur c'est l'erreur de compensation de mouvement entre les deux images clés qui est analysée. Selon les résultats des simulations montrés dans [TTT⁺06b], la solution permet d'améliorer les performances débit-distorsion jusqu'à 5 dB pour certaines séquences vidéo à résolution spatiale QCIF (176x144 pixels) et à 30 images par seconde.

Une approche similaire est présentée dans la section 4.7 de cette thèse.

2.3.3.8 Utilisation de la voie de retour

La voie de retour est une contrainte de l'architecture de Stanford, car sa présence implique que le codeur et le décodeur doivent travailler en même temps et en temps

réel, c'est-à-dire "en-ligne". De plus, son utilisation introduit un retard important dans le système.

Des efforts pour réduire l'utilisation de la voie de retour ont été fait. Dans [BP07], Brites *et al.* proposèrent une solution qui permet de la supprimer complètement (le contrôle de débit étant effectué au niveau du codeur par une estimation approximative de corrélation avec un algorithme d'estimation de mouvement simplifié), mais au prix d'une dégradation considérable des performances débit-distorsion du codeur (jusqu'à 1.2 dB moins bons qu'avec le contrôle de débit au décodeur). Néanmoins, cette solution peut être adoptée dans les cas où l'utilisation de la voie de retour n'est pas possible physiquement.

Un autre type de solution consiste à effectuer un contrôle hybride, c'est-à-dire partiellement au codeur, et partiellement au décodeur. Le codeur tâche de "deviner" le débit nécessaire, tandis que le décodeur peut toujours utiliser la voie de retour si ce débit n'est pas suffisant. Le but de ce type d'approche est de réduire l'usage de la voie de retour (et ainsi le retard du système), mais aussi la complexité du décodeur (car le nombre de requêtes des bits de parité est diminué). Une solution de ce type est proposée dans [AABP08], dont l'idée générale est d'analyser la statistique (le nombre de requêtes pendant le décodage de quelques images précédentes) pour prédire le nombre de requêtes pour l'image actuelle.

Enfin, une dernière solution, effectuant la prédiction sur la base d'un débit minimal théorique $H(X|Y)$, est présenté dans la section 4.4.1 de cette thèse.

2.3.3.9 Estimation de taux d'erreurs

L'élément le moins réaliste de l'architecture de Stanford [AZG02, GARRM05] est sans doute le fait que le taux d'erreur dans le plan de bits décodé était estimé en le comparant à celui du codeur. En effet, si ce plan de bits original est disponible au décodeur, il n'aurait pas besoin de le décoder.

Différentes solutions ont été proposées par Lajnef [Laj06] (adoptée dans DISCOVER) et Tagliasacchi *et al.* [TPPT07], utilisant toutes les deux le rapport de vraisemblance sur les bits décodés :

$$\Lambda_i = \log \frac{\Pr(x_i = 1)}{\Pr(x_i = 0)}, \quad (2.32)$$

où $\Pr(x_i = b)$ est la probabilité *a posteriori* que le bit x_i soit égal à b .

La solution de [Laj06] renforcée par la somme CRC est présentée dans la section 4.4.2.

2.3.3.10 Conclusion

Les techniques présentées ci-dessus améliorent de façon significative le codeur d'origine présenté dans [AZG02, ARSG04, GARRM05], notamment en ce qui concerne les performances débit-distorsion et les limitations pratiques (comme l'utilisation de

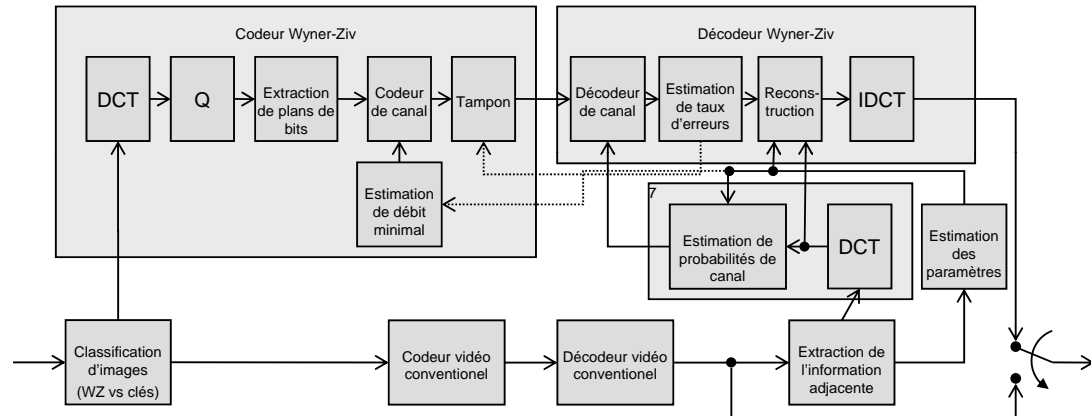


FIG. 2.18 – Le schéma du système de codage vidéo DISCOVER

voie de retour et la connaissance de l'image originale au décodeur). Dans la section suivante nous présentons de façon plus détaillée un système de codage réel incorporant certaines de ces techniques, et réalisant les meilleures performances dans le domaine de codage vidéo distribué [dis07].

2.3.4 Système de codage vidéo de Wyner-Ziv DISCOVER

Dans cette section, nous présentons le système de codage vidéo de Wyner-Ziv, qui a été développé dans le projet DISCOVER [dis, dis07]. Il est basé sur l'architecture de Stanford dans le domaine transformé [ARSG04, GARRM05]. Le schéma général est présenté sur la figure 2.18, où la plupart de blocs ont le même sens que les blocs correspondants de l'architecture de base montrée sur la figure 2.17. Ci-dessous nous décrivons les outils principaux du système DISCOVER qui le différencie de l'architecture de base. Le lecteur peut aussi se référer à [AAD⁺07] pour une description globale du codeur DISCOVER.

2.3.4.1 Classification d'images

Le codeur DISCOVER comporte deux modes concernant la taille de GOP : les GOPs de taille fixe, et les GOPs de taille variable. Dans le deuxième cas, l'algorithme proposé dans [ABP06] est utilisé. Cette approche adaptative est constituée en deux étapes :

1. **Estimation d'activité** - Entre deux images consécutives numérotées i et j ($|i - j| = 1$) les 4 mesures d'activité (de changement) suivantes sont calculées :

- Différence d'histogrammes :

$$DH(i, j) = \frac{1}{D_f} \sum_{k=0}^L |h_i(k) - h_j(k)|, \quad (2.33)$$

- Histogramme de différences :

$$HD(i, j) = \frac{1}{D_f} \left(\sum_{k=0}^{L/2-\alpha} h_{i-j}(k) + \sum_{k=L/2+\alpha}^L h_{i-j}(k) \right), \quad (2.34)$$

- Histogramme de différences par blocs :

$$BHD(i, j) = \sum_{b=0}^{N_b} \sum_{k=0}^L |h_i(b, k) - h_j(b, k)|, \quad (2.35)$$

- Et différence de variance par blocs :

$$BVD(i, j) = \sum_{b=0}^{N_b} |\sigma_i^2(b) - \sigma_j^2(b)|, \quad (2.36)$$

où h_i est l'opérateur d'histogramme à L niveaux appliqué sur les valeurs de luminance de pixels de l'image i , h_{i-j} est l'opérateur d'histogramme appliqué sur la différence d'images i et j dans le domaine pixel, D_f est le nombre total de pixels en images, N_b est le nombre de blocs (de taille 8x8), et $\sigma_i^2(b)$ est la variance de luminance dans le bloc b . Dans [ABP06] Ascenso *et al.* proposent les valeurs suivantes pour les constantes : $L = 32$ pour DH , $L = 64$ et $\alpha = 16$ pour HD , $L = 8$ pour BHD .

- Formation de GOP** - Les GOP sont ensuite formés par l'algorithme de groupement présenté ci-dessous (avec une contrainte M sur la taille maximale de GOP) :
 - $N_c = M - 1$ vecteurs de dimension 4 sont formés (pour chaque paire d'images adjacentes dans l'ensemble de M images) :

$$x_i = [DH(i-1, i) \quad HD(i-1, i) \quad BHD(i-1, i) \quad BVD(i-1, i)], \quad i = 1, \dots, M-1, \quad (2.37)$$

et normalisés :

$$x_i = \frac{N_c x_i}{\sum_{j=0}^{N_c} x_j}. \quad (2.38)$$

- Pour le groupe d'images courant, accumulation du mouvement :

$$y_i = x_{i-1} + x_i, \quad i = 2, \dots, N_c. \quad (2.39)$$

- Définition de l'indice c correspondant au minimum du mouvement accumulé :

$$c = \arg \min_{1 \leq i \leq N_c} \|y_i\|. \quad (2.40)$$

- Groupement des images correspondantes et modification des métriques (2.38) comme suit :

$$x_{i-1} = \begin{cases} x_i, & i > c, \\ y_i, & i = c \end{cases} \quad c \leq i \leq N_c. \quad (2.41)$$

- $N_c = N_c - 1$ et retour à l'étape 2 sauf si la condition suivante est vérifiée :

$$y_c > \phi \vee N_c = 1. \quad (2.42)$$

A chaque itération, l'algorithme groupe dans le même GOP les deux images adjacentes les plus proches en terme de métrique (2.38) parmi les M images analysées. Le paramètre ϕ définit le critère d'arrêt : plus sa valeur est plus élevée, plus les GOP sont plus petits et homogènes (par le contenu). L'utilisation de cet algorithme permet d'avoir des GOP grands pour les segments statiques ou avec peu de mouvement, et des GOP courts pour les segments avec un mouvement fort. Par conséquent, les performances débit-distorsion sont améliorées par rapport à l'utilisation des GOPs de taille fixe.

2.3.4.2 Transformée et quantification

Dans DISCOVER, une transformée analogue à la DCT par blocs 4x4 en nombres entiers [MHKK03] est utilisée. Les nombres de niveaux de quantification de chacune des 16 bandes de fréquence sont définis dans une matrice dite de quantification, fixée pour toute la séquence selon la distorsion voulue pour les images WZ. Ces matrices données dans [BAP06a] sont présentées dans le tableau 2.5 (les valeurs correspondent aux positions de coefficients DCT 4x4, avec le coefficient à basse fréquence (appelé DC) dans le coin en haut à gauche). Si le nombre de niveaux est nul, aucun plan de bits n'est transmis pour la bande correspondante, et pendant la reconstruction, on recopie l'information adjacente.

Le processus de quantification est différent pour la bande à basse fréquence (appelée "DC" en anglais de "Direct Current") et les autres bandes à hautes fréquences ("AC" de "Alternating Current"). Pour la bande DC (dont les valeurs non-négatives sont limitées par $MaxVal_{DC}$), un quantificateur scalaire uniforme est utilisé :

$$x^q = q, \quad \text{si } x \in [qW_{DC}, (q+1)W_{DC}), \quad (2.43)$$

où q est l'indice de quantification, W_{DC} est le pas de quantification défini par :

$$W_{DC} = \left\lceil \frac{MaxVal_{DC}}{2^{M_{DC}}} \right\rceil, \quad (2.44)$$

avec $2^{M_{DC}} = 2^{M_1}$ le nombre de niveaux de quantification réservés à la bande DC (selon la matrice choisie).

TAB. 2.5 – Matrices de quantification de DISCOVER

(a) QIndex 1				(b) QIndex 2				(c) QIndex 3				(d) QIndex 4			
16	8	0	0	32	8	0	0	32	8	4	0	32	16	8	4
8	0	0	0	8	0	0	0	8	4	0	0	16	8	4	0
0	0	0	0	0	0	0	0	4	0	0	0	8	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0

(e) QIndex 5				(f) QIndex 6				(g) QIndex 7				(h) QIndex 8			
32	16	8	4	64	16	8	8	64	32	16	8	128	64	32	16
16	8	4	4	16	8	8	4	32	16	8	4	64	32	16	8
8	4	4	0	8	8	4	4	16	8	4	4	32	16	8	4
4	4	0	0	8	4	4	0	8	4	4	0	16	8	4	0

Pour les autres bandes AC ($b > 1$), dont l'intervalle dynamique des valeurs est $[-MaxVal_b, MaxVal_b]$, un quantificateur scalaire à zone morte est utilisé :

$$x^q = \begin{cases} q, & \text{si } (x \geq W_b \& x \in [qW_b, (q+1)W_b)) \vee (x < -W_b \& x \in [(q-1)W_b, qW_b)), \\ 0, & \text{si } x \in [-W_b, W_b), \end{cases} \quad (\text{la taille de l'intervalle } 0 \text{ est double}) \quad (2.45)$$

avec le pas de quantification W_b donné par :

$$W_b = \left\lceil \frac{2MaxVal_b}{2^{M_b}} \right\rceil = \left\lceil \frac{MaxVal_b}{2^{M_b-1}} \right\rceil, \quad (2.46)$$

où 2^{M_b} est le nombre de niveaux de quantification pour la bande b .

2.3.4.3 Le codage par plan de bits

La dernière version du codeur DISCOVER adopte les codes LDPC pour le codage de Slepian-Wolf de plans de bits. Au début du projet le codeur utilisait les codes turbo ; ils ont été remplacés par les codes LDPC parce que ceux-ci réalisent de meilleures performances dans le contexte de codage vidéo distribué [VAG05, VAG06]. Le codage et décodage de séquences binaires avec les codes LDPC en présence de l'information adjacente ont déjà été abordés dans la section 2.1.2.5. Dans cette sous-section nous ne précisons que quelques détails supplémentaires de l'implémentation.

Le codage se fait avec une matrice \mathbf{H} de parité. Les codes réguliers (voir la section 1.3.3) sont utilisés, dont la sortie est concaténée avec un accumulateur [VAG05], ce qui permet de faire une adaptation de débit : en transmettant progressivement les syndromes accumulés selon les requêtes du décodeur. Après le calcul, les syndromes accumulés sont mis dans un tampon temporaire comme indiqué sur la figure 2.18.

Le décodage est réalisé avec l'algorithme de propagation des croyances décrit dans la section 2.1.2.5, les probabilités *a priori* des bits étant estimées selon la formule (2.29).

L'estimation des paramètres de corrélation α_k est décrite ci-dessous, dans la section 2.3.4.6.

Le nombre maximal d'itérations de l'algorithme de propagation de croyances est fixé à 100. Après chaque itération, le nombre d'équations de parité non-vérifiées est calculé. S'il est non-nul et le nombre maximal d'itérations n'est pas atteint, l'itération suivante est lancée. Dans le cas où le nombre maximal d'itérations est atteint, le décodage est déclaré faux, et des syndromes accumulés supplémentaires sont demandés au codeur. Si le nombre d'équations de parité non-vérifiées est nul, le contrôle de décodage est effectué par vérification de la somme CRC du résultat (en la comparant avec la somme CRC reçue du codeur). Dans le cas où la somme CRC calculée ne correspond pas à celle transmise par le codeur, une nouvelle itération est lancée, ou, si le nombre maximal d'itérations est atteint, des syndromes accumulés supplémentaires sont demandés. Dans le cas contraire, le décodage est déclaré fini avec succès.

Le décodage avec tous les syndromes accumulés transmis est un cas particulier. Cela peut arriver lors du décodage de plans de bits les moins importants si la corrélation avec l'information adjacente n'est pas suffisante. Le taux de compression dans ce cas est 1 (pour chaque bit codé 1 bit de syndrome est transmis). Bien que le décodage itératif avec l'algorithme de propagation des croyances est toujours possible, le décodage direct en utilisant l'inverse de la matrice de parité \mathbf{H}^{-1} est plus efficace et sûr. Cette possibilité d'un décodage direct quand le taux est égal à 1 constitue un avantage important des codes LDPC par rapport aux codes turbo. En effet, le taux d'un code turbo peut atteindre des valeurs plus élevées (plus d'un bit de parité transmis pour 1 bit codé) si la corrélation est très basse.

2.3.4.4 Estimation de débit minimal

Pour réduire le nombre de requêtes pendant le décodage de plans de bits, un module d'estimation de débit minimal théorique est introduit dans le codeur. Le nombre de syndromes accumulés, correspondant au débit estimé, est transmis au décodeur évitant ainsi au décodeur d'avoir à les demander et à faire des essais de décodage inutiles.

Le débit minimal théorique peut être déterminé par la fonction débit-distorsion de Wyner-Ziv (pour le bruit Gaussien à variance σ^2) :

$$R_{WZ}(D) = \frac{1}{2} \log_2 \frac{\sigma^2}{D} \quad (2.47)$$

pour une certaine distorsion D (voir la section 1.4.2). La distorsion associée à la transmission de $0 \leq b \leq M_k$ plans de bits d'un coefficient x_i peut être estimée comme suit :

$$\hat{D}_b(x_i) = \int_{-\infty}^{+\infty} (x_i - f(n, x_i))^2 p_z(n) dn, \quad (2.48)$$

où $p_z(n)$ est la densité de probabilité du bruit de corrélation, et $f(n, x_i)$ modélise la fonction de reconstruction utilisée par le décodeur.

Dans [Laj06] une approximation a été proposée pour éviter à calculer cette intégrale numériquement. Pour simplifier le calcul, une reconstruction sous-optimale est supposée, à savoir :

$$f(n, x_i) = \begin{cases} l & \text{si } n + x_i < l, \\ x_i + n & \text{si } l \leq n + x_i \leq u, \\ u & \text{si } n + x_i > u, \end{cases} \quad (2.49)$$

où $x_i + n$ modélise la valeur de l'information adjacente y_i , $[l, u]$ est l'intervalle de quantification pour x_i , défini avec les b bits les plus importants de l'indice de quantification x_i^q (si $b = 0$, $[l, u]$ couvre toutes les valeurs possibles de X).

Ainsi, pour un modèle de corrélation Laplacien $p(n) = \frac{\alpha}{2} e^{-\alpha|n|}$ en utilisant (2.49) la distorsion (2.48) peut être écrite comme :

$$\hat{D}_b(x_i) = (x_i - l)^2 \int_{-\infty}^{l-x_i} \frac{\alpha}{2} e^{-\alpha|n|} dn + \int_{l-x_i}^{u-x_i} \frac{\alpha}{2} n^2 e^{-\alpha|n|} dn + (x_i - u)^2 \int_{u-x_i}^{+\infty} \frac{\alpha}{2} e^{-\alpha|n|} dn. \quad (2.50)$$

Après l'intégration on obtient :

$$\hat{D}_b(x_i) = \frac{2 - [1 + \alpha(x_i - l)] e^{-\alpha(x_i - l)} - [1 + \alpha(u - x_i)] e^{-\alpha(u - x_i)}}{\alpha^2}. \quad (2.51)$$

Dans cette expression, le paramètre α est estimé par le décodeur (voir la section 2.3.4.6 ci-dessous) et transmis par la voie de retour.

La distorsion totale pour la séquence X est donnée par :

$$\hat{D}_b = \sum_i \hat{D}_b(x_i). \quad (2.52)$$

Sachant que la transmission d'un plan de bits b permet de diminuer la distorsion de reconstruction de D_{b-1} jusqu'à D_b , le débit minimal théorique pour le plan de bits b est ensuite défini par :

$$\hat{R}_{min}^b = R_{WZ}(\hat{D}_b) - R_{WZ}(\hat{D}_{b-1}) = \frac{1}{2} \log_2 \frac{\sigma^2}{\hat{D}_b} - \frac{1}{2} \log_2 \frac{\sigma^2}{\hat{D}_{b-1}} = \frac{1}{2} \log_2 \frac{\hat{D}_{b-1}}{\hat{D}_b}, \quad (2.53)$$

en posant $\hat{D}_{-1} = \sigma^2$.

Une autre méthode d'estimation de débit minimal théorique utilisant la borne de Slepian-Wolf pour des séquences binaires, est proposée dans la section 4.4.1 de cette thèse. Ainsi dans cette méthode on est dispensée d'avoir à supposer une méthode (optimale ou sous-optimale) de reconstruction au décodeur.

2.3.4.5 Génération de l'information adjacente

L'algorithme de génération de l'information adjacente est présenté en détails dans [ABP05a] pour des GOP de taille 2, et généralisé pour les autres tailles dans [ABP06]. Le schéma de l'algorithme est présenté sur la figure 2.19. Il comporte les étapes suivantes :

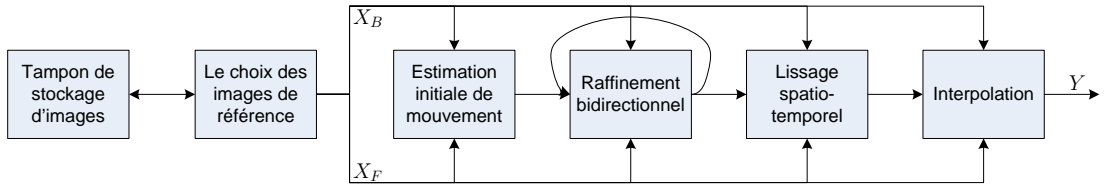


FIG. 2.19 – Le schéma de l'extraction de l'information adjacente dans DISCOVER

1. **Choix des images de référence X_B et X_F .** Pour les GOP de taille 2, ce sont les images clés adjacentes à l'image WZ courante. Pour les GOP de taille $n > 2$, les images WZ dans le GOP sont décodées suivant un ordre prédéfini comme indiqué dans [ABP06] (les images plus proches du milieu du GOP sont décodées les premières). Dans ce cas, les images X_B et X_F sont les deux images les plus proches de l'image WZ courante parmi les images clés adjacentes au GOP et les images WZ déjà décodées à l'intérieur du GOP. Les images ainsi choisies sont extraites du tampon de stockage d'images décodées.
2. **Estimation de mouvement entre X_B et X_F .** Les images X_B et X_F sont d'abord traitées avec un filtre passe-bas pour améliorer la fiabilité de l'estimation contre le bruit. Ensuite, la recherche exhaustive de vecteurs de mouvement est effectuée avec un critère à minimiser défini dans [ABP06] comme :

$$WMAD(d_x, d_y) = \left(1 + k \sqrt{d_x^2 + d_y^2}\right) \frac{1}{|B_i|} \sum_{(x,y) \in B_i} |X_F(x, y) - X_B(x + d_x, y + d_y)|, \quad (2.54)$$

où $\mathbf{d} = (d_x, d_y)$ est le vecteur testé, B_i est le bloc courant (de taille 16x16 pixels), et k (fixé à 0.05 dans [ABP06]) est le coefficient de lissage qui pénalise les vecteurs trop éloignés de l'origine (0,0). Le lissage permet d'avoir de bonnes performances même avec une fenêtre de recherche assez large, e.g. ± 32 pixels.

3. **Raffinement bidirectionnel des vecteurs initiaux.** Les vecteurs obtenus à l'étape précédente sont associés à la grille des blocs non-superposés de l'image X_F . Cependant, pour faire l'interpolation d'une manière efficace et ne pas avoir besoin de traiter spécialement les zones non-couvertes ("trous") et les zones couvertes par plusieurs blocs ("superpositions"), des vecteurs bidirectionnels associés aux blocs non-superposés dans l'image WZ courante sont requis. Ils sont définis comme suit.

Dans un premier temps, la valeur initiale d'un vecteur bidirectionnel est calculée en utilisant le vecteur unidirectionnel $\mathbf{d} = (d_x, d_y)$ qui transperce le plan de l'image WZ courante le plus près du centre du bloc traité, parmi tous les vecteurs obtenus à la première étape. Le vecteur bidirectionnel $(\mathbf{u}_B^0, \mathbf{u}_F^0)$ est ensuite donné par :

$$\begin{cases} \mathbf{u}_B^0 = \frac{t_B}{t_B + t_F} \mathbf{d}, \\ \mathbf{u}_F^0 = -\frac{t_F}{t_B + t_F} \mathbf{d}, \end{cases} \quad (2.55)$$

où t_B et t_F sont les distances de l'image WZ courante aux images X_B et X_F respectivement (voir la figure 2.20). Par exemple, pour les GOP de taille 2 $t_B = t_F = 1$.

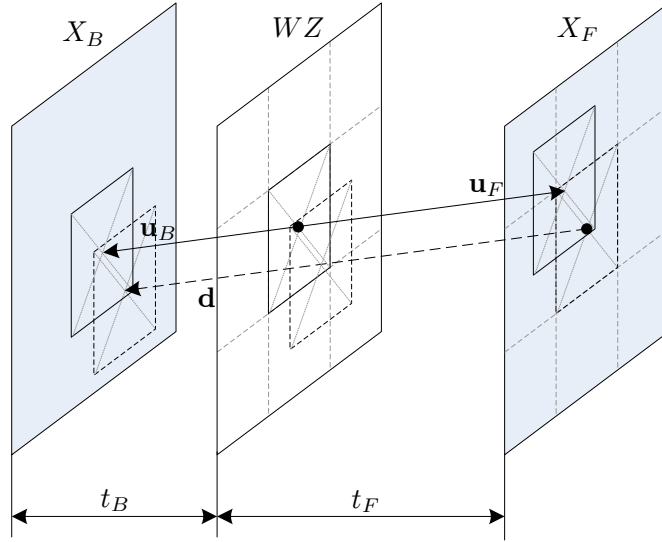


FIG. 2.20 – Obtention de vecteurs bidirectionnels à partir des vecteurs initiaux

Dans un deuxième temps, le vecteur $(\mathbf{u}_B^0, \mathbf{u}_F^0)$ est raffiné en cherchant le vecteur bidirectionnel $(\mathbf{u}_B, \mathbf{u}_F) = (\mathbf{u}_B^0 + \Delta, \mathbf{u}_F^0 - \Delta)$ qui minimise le critère suivant :

$$MAD(\mathbf{u}_B, \mathbf{u}_F) = \frac{1}{|B_i|} \sum_{s \in B_i} |X_F(s + \mathbf{u}_F) - X_B(s + \mathbf{u}_B)| \quad (2.56)$$

pour tout $\Delta \in (\pm f, \pm f)$, où f est la taille de la fenêtre de raffinement (en général pas très grande, 3 ou 4 pixels).

Pour améliorer la performance une approche hiérarchique a été proposée dans [ABP06] qui consiste à faire d'abord le raffinement pour des blocs de grande taille (16x16 pixels), et à déduire de ces valeurs raffinées les vecteurs des petits blocs (8x8).

4. **Lissage spatio-temporel du champ de mouvement.** Après le raffinement, le champ entier de mouvement est lissé par un filtre médian pondéré proposé dans

[ABB96]. Pour chaque bloc, un ensemble U de vecteurs-candidats est formé en ajoutant au vecteur courant bidirectionnel $\mathbf{u}^c = (\mathbf{u}_B^c, \mathbf{u}_F^c)$ du bloc traité les vecteurs $\mathbf{u}^i = (\mathbf{u}_B^i, \mathbf{u}_F^i)$ des blocs voisins. Le vecteur lissé $\mathbf{u}' = (\mathbf{u}'_B, \mathbf{u}'_F)$ est défini comme suit :

$$\mathbf{u}' = \arg \min_{\mathbf{u} \in U} \sum_{\mathbf{u}^i \in U} w_i \|\mathbf{u} - \mathbf{u}^i\|_2, \quad (2.57)$$

où w_i est le poids associé au vecteur \mathbf{u}^i :

$$w_i = \frac{MSE(\mathbf{u}^c)}{MSE(\mathbf{u}^i)}, \quad (2.58)$$

avec $MSE(\mathbf{u}) = MSE(\mathbf{u}_B, \mathbf{u}_F)$ étant l'équivalent de (2.56) dans la norme L_2 :

$$MSE(\mathbf{u}_B, \mathbf{u}_F) = \frac{1}{|B_i|} \sum_{\mathbf{s} \in B_i} [X_F(\mathbf{s} + \mathbf{u}_F) - X_B(\mathbf{s} + \mathbf{u}_B)]^2. \quad (2.59)$$

Les poids w_i sont choisis de telle façon que les vecteurs-candidats pertinents (dont l'erreur de compensation (2.59) est plus faible) ont plus d'influence sur le processus de lissage que les vecteurs-candidats non-pertinents.

5. **Interpolation.** La dernière étape de génération de l'information adjacente est constitué par l'interpolation bidimensionnelle utilisant le champ de mouvement lissé. Elle est effectuée via le calcul suivant :

$$Y(\mathbf{s}) = \frac{t_F X_B(\mathbf{s} + \mathbf{u}_B) + t_B X_F(\mathbf{s} + \mathbf{u}_F)}{t_F + t_B}, \quad (2.60)$$

où $(\mathbf{u}_B, \mathbf{u}_F)$ est le vecteur associé au bloc contenant le pixel $\mathbf{s} = (x, y)$.

2.3.4.6 Modélisation de la corrélation

Comme l'architecture de Stanford, le système DISCOVER adopte le modèle de corrélation Laplacien, dont les paramètres sont estimés pendant le décodage (mode dit "en-ligne"), et non pas présumés connus ou estimés préalablement comme dans l'architecture d'origine [AZG02, GARRM05] (mode dit "hors-ligne"). L'algorithme d'estimation ci-dessous est proposé par Brites et Pereira [BP08].

Les paramètres du modèle pour les différentes bandes de fréquence sont estimés indépendamment en utilisant la bande respective du résidu de compensation de mouvement. L'image du résidu est calculée comme suit :

$$R(\mathbf{s}) = \frac{X_B(\mathbf{s} + \mathbf{u}_B) - X_F(\mathbf{s} + \mathbf{u}_F)}{2}, \quad (2.61)$$

Ensuite, cette image R est transformée (via une DCT 4x4), et les 16 bandes de fréquence $R_k, k = 1, \dots, 16$, sont formées. Soit $r_{k,i}$ le coefficient à la position i dans la bande k . La variance $\sigma_{|k|}^2$ des valeurs absolues des coefficients de la bande k est donnée par :

$$\sigma_{|k|}^2 = \mathbb{E} [|r_{k,i}| - \mu_{|k|}]^2 = \frac{1}{N_k} \sum_i [|r_{k,i}| - \mu_{|k|}]^2, \quad (2.62)$$

où $\mu_{|k|} = \sum_i |r_{k,i}|$ est la moyenne des valeurs absolues de la bande R_k , et N_k est la taille de la bande. Tous les coefficients $r_{k,i}$ sont séparés en deux groupes selon leur valeur absolue : les coefficients "normaux" ($[|r_{k,i}| - \mu_{|k|}]^2 \leq \sigma_{|k|}^2$), et les coefficients "aberrants" ($[|r_{k,i}| - \mu_{|k|}]^2 > \sigma_{|k|}^2$). La raison de les séparer est la suivante.

Dans les positions où les coefficients du résidu suivent la statistique générale (les coefficients "normaux"), le modèle général caractérisé par un paramètre α_k est valable avec une probabilité importante. Parallèlement, dans les positions où les coefficients du résidu sont "aberrants", un modèle particulier doit être appliqué, avec un paramètre $\alpha_{k,i} \neq \alpha_k$. L'estimation de ces paramètres est réalisée avec la formule suivante :

$$\hat{\alpha}_{k,i} = \begin{cases} \hat{\alpha}_k = \sqrt{\frac{2}{\sigma_k^2}} & \text{if } [|r_{k,i}| - \mu_{|k|}]^2 \leq \sigma_{|k|}^2, \\ \sqrt{\frac{2}{[|r_{k,i}| - \mu_{|k|}]^2}} & \text{if } [|r_{k,i}| - \mu_{|k|}]^2 > \sigma_{|k|}^2, \end{cases} \quad (2.63)$$

où σ_k^2 est la variance des coefficients $r_{k,i}$:

$$\sigma_k^2 = \frac{1}{N_k} \sum_i \left[r_{k,i} - \frac{1}{N_k} \sum_j r_{k,j} \right]^2. \quad (2.64)$$

La méthode présentée permet une modélisation assez fine (au niveau des coefficients individuels si nécessaire) du bruit de corrélation. Son point faible est qu'elle suppose que le résidu R est une bonne approximation du bruit $Z = X - Y$. Et en pratique, cette hypothèse n'est pas toujours vérifiée en pratique.

2.3.4.7 Reconstruction

La reconstruction est faite selon la formule optimale (dans le sens de la distorsion quadratique minimale) pour un modèle de corrélation Laplacien :

$$\hat{x}_{k,i} = \begin{cases} l + \frac{1}{\hat{\alpha}_{k,i}} + \frac{\Delta}{1 - e^{-\hat{\alpha}_{k,i}\Delta}}, & y_{k,i} < l \\ y_{k,i} + \frac{(\gamma + \frac{1}{\hat{\alpha}_{k,i}})e^{-\hat{\alpha}_{k,i}\gamma} - (\delta + \frac{1}{\hat{\alpha}_{k,i}})e^{-\hat{\alpha}_{k,i}\delta}}{2 - (e^{-\hat{\alpha}_{k,i}\gamma} + e^{-\hat{\alpha}_{k,i}\delta})}, & y \in [l, u) \\ u - \frac{1}{\hat{\alpha}_{k,i}} - \frac{\Delta}{1 - e^{-\hat{\alpha}_{k,i}\Delta}}, & y_{k,i} \geq u \end{cases}, \quad (2.65)$$

où $\gamma = y_{k,i} - l$, $\delta = u - y_{k,i}$, $\Delta = u - l$, $y_{k,i}$ est l'information adjacente, et $[l, u)$ est l'intervalle de quantification de $x_{k,i}$ (connu après le décodage de canal de l'indice de quantification).

L'établissement de cette formule est présentée dans la section 4.3.

2.3.4.8 Résultats des simulations

Les résultats des différentes simulations réalisées avec le système de codage vidéo DISCOVER sont disponibles sur le site web [dis07]. Entre autres, les performances débit-distorsion, la complexité du codeur et du décodeur (le temps de codage et décodage), le débit envoyé par la voie de retour ont été testés. Il a été montré que le système DISCOVER a des meilleurs résultats en terme de débit-distorsion qu'un codeur H.264 dans les modes "intra" et "no motion" sur des séquences vidéo avec peu de mouvement. La complexité du codeur DISCOVER est moins importante que celle du codeur H.264. Le décodeur DISCOVER est au contraire beaucoup plus complexe que celui de H.264.

Les séquences dont le contenu est caractérisé par un mouvement agité, restent un cas difficile pour le système DISCOVER, surtout avec des GOP de taille élevée. De telles séquences peuvent être codées plus efficacement avec un codeur conventionnel comme H.264 dans le mode "intra" par exemple.

2.4 Résumé des contributions

La première contribution de cette thèse, à laquelle le chapitre 3 est consacré, est l'étude des méthodes d'extraction de l'information adjacente (IA) pour le codage vidéo distribué. Contrairement aux travaux de Li et Delp [LD05, LLD07], plus orientés vers les aspects théoriques, ici les aspects pratiques d'utilisation des méthodes d'extraction de l'IA sont analysés, c'est-à-dire les performances débit-distorsion du système entier de codage avec des données vidéo réelles sont mesurées afin de juger de leur efficacité. Quelques nouvelles méthodes prometteuses ont également été proposées dans ce chapitre.

Le chapitre suivant est consacré aux problèmes de codage vidéo distribué liés à la modélisation de la corrélation entre l'information codée et l'information adjacente. Sont évoquées des questions d'estimation et de raffinement des paramètres du modèle, d'estimation du débit minimal théorique, d'optimisation des paramètres de codage en terme débit-distorsion etc.

Le codage Wyner-Ziv des sources avec mémoire est abordé dans le chapitre 5. Différents outils y sont proposés qui améliorent l'efficacité de codage en exploitant la mémoire de la source, tels que la transformée par ondelettes orientées, l'utilisation de filtres de prédiction dans la boucle de codage, ou le codage des indices de quantification avec des codes de Huffman.

Les contributions sont résumées ci-dessous dans les trois sections suivantes.

2.4.1 Extraction de l'information adjacente

L'obtention d'une information adjacente de bonne qualité est l'un des enjeux les plus importants en codage vidéo distribué. Beaucoup d'efforts ont été faits par des chercheurs dans le développement de nouveaux algorithmes de génération, la plupart étant basés sur des méthodes d'estimation de mouvement. L'état de l'art dans le domaine de l'extraction de l'information adjacente est établi dans le début du chapitre 3.

Nous commençons par proposer une nouvelle approche à l'extraction de l'information adjacente basée sur la représentation de l'image par maillage déformable. L'algorithme de construction et de déformation du maillage est inspiré des travaux de thèse de Gwenaëlle Marquant [Mar00]. Ici, nous appliquons sa technique d'estimation de mouvement dans le cadre du codage vidéo distribué. Étant plus complexe que la méthode d'estimation par blocs, dans les codeurs conventionnels l'estimation par maillage est en général caractérisée par une meilleure qualité de compensation de mouvement. Cependant, ce comportement n'a pas été pleinement confirmé dans ce cadre précis d'utilisation, c'est-à-dire que l'estimation par maillage n'est globalement pas meilleure que l'estimation par blocs.

Parallèlement, il a été remarqué que localement l'une des deux méthodes était souvent meilleure que l'autre. Ainsi, deux modèles d'utilisation simultanée de deux informations adjacentes ont été considérés.

Le premier modèle développé consiste à construire une seule information adjacente à partir de plusieurs disponibles. Cette méthode a été appelée "méthode hybride". L'idée générale est de combiner deux (ou plusieurs) séquences de symboles Y_1 et Y_2 en une seule séquence Y selon un critère de choix. Toutefois, la construction du critère idéal nécessite la connaissance de la séquence d'origine X , et par conséquent n'est pas envisageable. Un critère sous-optimal a été donc proposé.

La deuxième approche s'appelle "le décodage multi-hypothèse", et est déjà présent dans la littérature (par exemple [MKR05]). Elle consiste à utiliser deux séquences d'information adjacente Y_1 et Y_2 en même temps, en supposant une loi de distribution de X conditionnelle sachant Y_1 et Y_2 . Nous avons proposé une approximation de cette loi jointe, puis cette approche a été intégrée au décodeur DISCOVER, et les performances débit-distorsion ont été évaluées et comparées avec celles de la première approche dite "hybride".

La section suivante est consacrée à la construction et l'utilisation de l'information adjacente spatiale. Nous considérons un schéma dans lequel l'information adjacente spatiale est la version réduite en taille et quantifiée de la séquence d'origine X . Plus précisément, dans notre schéma c'est l'image WZ réduite et codée avec le codeur H.264 dans le mode "intra" en bas débit. Nous testons des paramètres différents de codage de cette information adjacente spatiale (le facteur de réduction, le paramètre de quantification), et montrons que l'information adjacente temporelle est presque toujours préférable, bien qu'elle peut avoir des valeurs de PSNR moins élevées. Cela nous

conduit à proposer dans cette section une mesure de qualité de l'information adjacente (appelée SIQ, "Side Information Quality") plus pertinente que la mesure PSNR pour évaluer les performances en terme débit-distorsion d'un codeur vidéo distribué.

Comme l'information de parité générée par le codeur de Slepian-Wolf cherche à corriger le bruit de corrélation entre l'information adjacente Y et l'image originale X , il semblait naturel de chercher à évaluer l'apport de méthodes de débruitage appliquées à l'information adjacente Y . Deux méthodes de débruitage ont été testées :

- la méthode de débruitage par seuillage des coefficients d'ondelettes orientées [CG06],
- et la méthode utilisant une transformée par des supports adaptatifs (plutôt que par blocs réguliers) [FKE07, DFKE07].

Ces méthodes ne sont pas du tout adaptées au type de bruit observé dans les images de l'information adjacente, ce qui explique leur basse efficacité dans ce cas-là (par rapport au cas avec le bruit synthétique ajouté). Cependant, la deuxième approche a permis d'améliorer légèrement les performances débit-distorsion du système.

2.4.2 Problèmes liés à la modélisation de la corrélation

Le chapitre 4 est consacré aux problèmes liés à la modélisation de la corrélation dans le codage vidéo distribué. Contrairement aux études théoriques sur le codage de sources distribuées, où le modèle de la source est supposé connu, les données vidéo sont extrêmement difficiles à modéliser. Par conséquent, la modélisation de la corrélation entre les images à coder et l'information adjacente est aussi un problème difficile.

Les contributions du chapitre 4 peuvent être séparées en trois parties. La première partie contient les approches qui traitent des problèmes au décodeur. Dans un premier temps le modèle de corrélation utilisé dans le système DISCOVER est étudié (proposé par Brites *et al.* [BP08]), ses points faibles sont identifiés, et leurs effets sur les performances débit-distorsion sont évalués. Ensuite, un algorithme est proposé dont le but est d'améliorer la précision de l'estimation des paramètres de corrélation dans un système réel de codage vidéo. Cet algorithme utilise les principes de l'algorithme EM (Expectation-Maximization) [Moo96] pour raffiner les paramètres du modèle au fur et à mesure du décodage turbo, en utilisant les probabilités *a posteriori* dites souples des décodeurs SISO constituant le décodeur turbo.

La reconstruction optimale en terme de l'erreur quadratique moyenne est également abordée dans la première partie du chapitre. Les formules qui permettent le calcul direct des valeurs ayant la distorsion quadratique attendue minimale sont établies pour un modèle de corrélation Laplacien. Elles sont également étendues au décodage multi-hypothèse décrite dans le chapitre 3. Ce travail a été publié dans l'article [KNG07], qui a été récompensé par le prix du meilleur article étudiant ("Best Student Paper Award") de la conférence.

La deuxième partie du chapitre propose une solution dite "hybride" de contrôle

du débit. Il s'agit d'une solution ayant deux modules, l'un situé à l'encodeur, l'autre au décodeur. Le module situé à l'encodeur estime le débit minimal théorique qui sera nécessaire au décodage d'une séquence de bits, connaissant les paramètres du modèle de corrélation qui sont envoyés depuis le décodeur par la voie de retour. Ce débit estimé constitue la quantité de bits de parité avec laquelle le décodeur commence le décodage. Par conséquent, le décodeur ne doit demander que les bits de parité restants (dû à la différence entre le débit minimal théorique et le débit réel), et sa complexité est réduite de façon significative. Le module situé au décodeur estime le taux d'erreur à l'issue du décodeur turbo. Si ce taux est élevé, une requête supplémentaire de bits de parité est faite, et le décodage turbo est relancé. Le résultat est ensuite vérifié en utilisant une somme de parité (CRC) de la séquence codée, qui est envoyée aussi par le codeur.

Ensuite, nous abordons les problèmes liés aux contrôle de débit au codeur, qui nécessite une modélisation de la corrélation du côté du codeur. Les modifications qui sont proposées dans cette troisième partie du chapitre, sortent du cadre de codage purement distribué car elles nécessitent un traitement inter-images (les images de la vidéo ne sont plus considérées par l'encodeur comme indépendantes). Mais elles permettent ainsi d'atteindre des performances débit-distorsion compétitives par rapport à celles des systèmes de codage vidéo conventionnels.

Dans un premier temps, nous proposons un algorithme pour définir de façon adaptative la matrice de quantification pour les images Wyner-Ziv, au lieu de fixer cette matrice pour la séquence entière comme le fait le codeur DISCOVER. Cela permet de supprimer l'étape préliminaire de codage qui consiste à trouver une correspondance entre les paramètres de codage des images clés et des images WZ respectivement, obtenant ainsi une qualité comparable en moyenne pour les deux types d'images. Notons que cette correspondance doit être établie préalablement pour une séquence vidéo que l'on veut coder, et qu'une correspondance définie pour toute la séquence ne garantit pas que deux images adjacentes vont avoir une qualité comparable. La méthode proposée permet de contrôler la qualité des images WZ de façon relativement robuste. En outre, la matrice de quantification est optimisée du point de vue débit-distorsion, ce qui permet d'avoir de meilleures performances débit-distorsion pour la séquence entière.

L'approche d'optimisation débit-distorsion nécessite une estimation fiable des paramètres du modèle de corrélation. C'est pourquoi nous avons développé une méthode robuste d'estimation des paramètres au codeur. Cette méthode consiste à faire la construction complète de l'information adjacente pour une petite partie de l'image, contenant de 10 à 20% du nombre total de blocs. Pour que l'estimation basée sur ces échantillons soit pertinente, les blocs sont choisis aléatoirement. Les résultats expérimentaux montrent que l'estimation avec la méthode proposée est beaucoup plus fiable qu'avec la méthode de Brites *et al.* [BP08] (employée par le décodeur), cependant au détriment de la complexité du codeur qui se trouve augmentée.

La dernière contribution est orientée vers une exploitation plus efficace de la corrélation entre les images par le codeur. Il s'agit d'un changement adaptatif du mode de codage, en ajoutant la possibilité de choix du mode "intra" pour une partie d'une image WZ. Contrairement à l'approche de Ascenso *et al.* [ABP06], le choix est fait au niveau des blocs de l'image WZ, et non pas au niveau des images entières. Cette approche est inspirée du codeur PRISM [PR02, PR03b, PMR07]. Ainsi, une technique simple de détection de mouvement est à la base du critère de décision. Les blocs qui correspondent à la partie de l'image avec mouvement détecté, sont codés avec le codeur H.264 en mode "intra", tandis que les autres blocs sont codés en mode WZ. Le masque de blocs sélectionnés pour le codage en mode "intra" est codé par un codeur arithmétique.

Les résultats expérimentaux montrent que les performances débit-distorsion sont améliorées de façon significative dans certains cas. De plus, la complexité moyenne du décodeur est réduite due au fait que la corrélation avec l'information adjacente pour les données codées en mode WZ est plus élevée.

2.4.3 Codage de Wyner-Ziv des sources avec mémoire

Dans le chapitre 5 nous abordons le problème du codage Wyner-Ziv des sources avec mémoire. Par la mémoire dans les données vidéo nous sous-entendons la corrélation spatiale à l'intérieur d'une image, car les images sont considérées par un codeur Wyner-Ziv comme des sources différentes, et sont codées séparément.

Dans la plupart des systèmes actuels le problème de corrélation spatiale dans les images WZ est traité avec une transformée de type DCT. Cependant, il existe des transformées plus efficaces en terme de compaction de l'énergie que la DCT, e.g. les transformées par ondelettes. Ainsi dans notre première contribution nous intégrons au codeur DISCOVER la transformée par ondelettes orientées [CG06] au lieu de la DCT. Cette transformée est particulièrement efficace sur les contours dans une image, car elle permet une analyse en directions diagonales, et non pas seulement horizontales ou verticales (comme la DCT). Cela permet d'obtenir des gains importants en terme de performances débit-distorsion pour des séquences vidéo qui contiennent beaucoup de contours.

Nous traitons la mémoire au niveau d'une bande de fréquence. Dans le codage conventionnel, ce problème se résout souvent par un algorithme de prédiction DPCM ("Differential Pulse Code Modulation"). Le DPCM pour une source X avec mémoire consiste à construire un filtre minimisant la variance de l'erreur de prédiction d'un symbole X_t à partir des symboles précédents X_{t-1}, X_{t-2} etc. Dans le codage Wyner-Ziv outre la variance de l'erreur de prédiction, la corrélation entre X et Y influence également sur les performances du système. Par conséquent, les deux paramètres doivent être pris en compte lors de la construction des filtres de prédiction pour X et Y . Nous proposons un algorithme de construction de filtres que nous appelons "WZ-DPCM".

Ensuite nous abordons le problème de mémoire au niveau des indices de quantification. Le codeur DISCOVER utilise un code à longueur fixe pour les représenter en binaire, ce qui est sous-optimal si la loi de distribution des indices n'est pas uniforme. Nous proposons d'utiliser un code de Huffman au lieu d'un code à longueur fixe. La loi de distribution des indices dans une bande de fréquence d'une image WZ nécessaire pour la construction d'un code de Huffman au décodeur peut être approchée par celle d'une image clé adjacente compte tenu de la forte corrélation entre les contenus des deux images adjacentes. Nous considérons également le problème de synchronisation des plans de bits qui apparaît dès que ceux-ci n'ont pas la même taille.

Enfin, la mémoire au niveau des plans bits est étudiée. Nous mettons en œuvre une technique qui permet de prendre en compte la mémoire dans l'algorithme BCJR utilisé dans le décodage turbo, en supposant un canal binaire symétrique (CBS) de transmission. Nous montrons que cette technique n'améliore pas les performances de DISCOVER. Cette observation confirme que le modèle de corrélation adopté au niveau des plans de bits (le modèle CBS) ne correspond pas aux données réelles.

2.5 Conclusion

Dans ce chapitre nous avons établi l'état de l'art du codage de sources distribuées en portant l'accent sur son application dans le domaine vidéo. Deux architectures populaires du codage vidéo distribué ont été présentées, notamment celle de Stanford et PRISM. Le codeur réel DISCOVER a été ensuite décrit avec beaucoup de détails, car il est utilisé comme plateforme de test pour les outils proposés dans cette thèse. Un résumé des contributions a été également dressé.

Deuxième partie

Contributions

Chapter 3

Side information extraction

In this chapter we study the problem of side information generation (or extraction) in distributed video coding (DVC). We start by explaining the importance of the side information, and introducing some notation that will be used below. This part is followed by a short review of existing state-of-the-art methods in this field. Finally, our contributions are presented and evaluated.

The quality of the side information (SI) is crucial for a Wyner-Ziv video codec. Talking about the side information quality, one should think about the correlation between it and the original frame, or, in other words, how “close” is the side information to the original frame. In video and image processing community, it is very common to use peak signal-to-noise ratio (PSNR) to measure how close one image is to another. It remains the most used objective image quality measure, although its correlation with the human perception is contested by many authors proposing new measures, taking the human visual system into account (e.g. [WBSS04]).

In distributed video coding it is not the human but the decoder who utilises the SI, the measure therefore should be adapted to the decoder’s “perception” of errors. In Section 3.4.4 we propose such a measure, which goal is to better evaluate the performance of denoising methods applied to the SI. However, the most reliable way to measure the quality of the SI, is to measure the overall rate-distortion performance of the codec integrating this side information; and this measure will be used most of the time in this chapter (sometimes along with other measures).

The SI quality is crucial for the rate-distortion performance, because it influences both the rate and the distortion in a Wyner-Ziv coding architecture with a return channel (better SI reduces both rate and distortion). Indeed, the transmission of quantisation indices is supposed to be errorless using channel codes (turbo codes or LDPC), and the parity bits rate required for a successful transmission, depends on the SI quality. On the other hand, the distortion is fully determined by the reconstruction procedure, which can be seen as the inverse quantisation using quantisation indices (supposedly transmitted without any error) along with the SI; thus the distortion is only determined by the SI quality. Consequently, if the quality of the SI decreases, both the

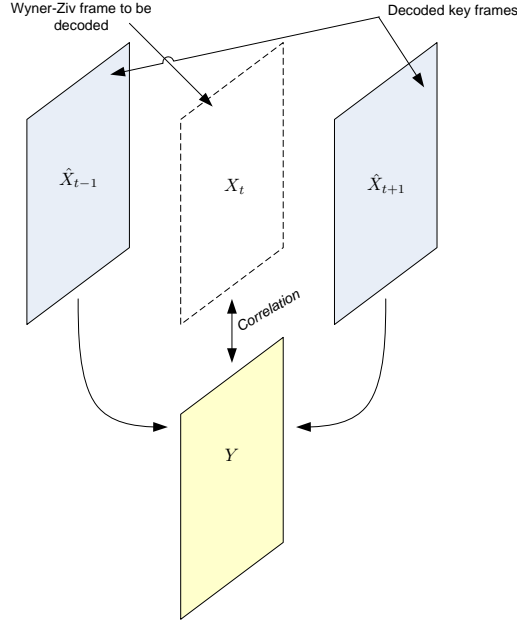


Figure 3.1: Side information construction scheme (GOP of size 2)

rate and the reconstruction distortion increase, penalising the codec's performance.

The side information construction task for a GOP of size 2 is illustrated on Figure 3.1. The two adjacent decoded key frames \hat{X}_{t-1} and \hat{X}_{t+1} are used to produce the estimate Y of the frame X_t . One of two frames \hat{X}_{t-1} or \hat{X}_{t+1} is often selected as a base frame for motion estimation, while the other is used as the reference to find the displacement between the two (see Section 2.3.4.5 for more details). That is why we use the notation $I_{base}(x, y)$ and $I_{ref}(x, y)$ for them, while denoting Y with $I_{ref}(x, y)$, and X with $I_{wz}(x, y)$. We use or omit pixel coordinates (x, y) depending on the context. We note that when the GOP size is larger than 2 frames, the two frames $I_{base}(x, y)$ and $I_{ref}(x, y)$, which are used to predict the frame $I_{wz}(x, y)$, might not be key frames anymore, but previously decoded Wyner-Ziv frames (more information about SI extraction with GOP sizes greater than 2, can be found in [ABP06]).

The SI extraction problem is very similar to the frame interpolation problem in frame rate conversion (FRC) applications, when it is required to increase the frame rate. However, most of FRC methods are oriented on the visual quality criterion, while in the DVC case the fidelity of the frame reconstruction is more important than the visual quality. This distinguishes the two groups of methods.

The simplest SI extraction methods are the frame repetition and the frame averaging (a usage of the frame averaging method is considered in [AZG02] for example).

The two methods are given by (3.1) and (3.2) respectively:

$$I_{si_rep}(x, y) = I_{base}(x, y), \quad (3.1)$$

$$I_{si_avg}(x, y) = \frac{1}{2} (I_{base}(x, y) + I_{ref}(x, y)). \quad (3.2)$$

Their advantage is the implementation simplicity and low computational complexity, but they fail to predict the image $I_{wz}(x, y)$ when there is motion in the video sequence.

Motion compensated methods, providing tremendous improvements over the simplest methods in the presence of motion, have rapidly become popular in conventional video coding, in spite of their relatively high computational complexity. However, in contrast to the classical predictive coding scheme, where the motion estimator has access to the frame to be motion compensated, in distributed video coding the motion estimation problem becomes ill-posed due to the fact, that the original frame is not known to the estimator. This ill-posedness is usually overcome by assuming the motion uniformity, i.e. that all objects (and possibly the camera) were moving along linear trajectories, and with a constant velocity, between the two time instants when the frames were captured.

Among numerous approaches and motion models, block-based motion compensation techniques offer some trade-off between the efficiency and the complexity. A detailed description of the block-based motion compensated SI extraction technique adopted in DISCOVER (proposed in [ABP05a]) can be found in Section 2.3.4.5.

The choice of the block size has critical effect on the performance of a block-based motion estimation method in general, and even more effect on the performance of a SI extraction method. To deal with this problem, adaptive block size selection algorithms were recently proposed in several works (e.g. [ATBS07, AP08]).

It is obvious that the efficiency of the motion compensated interpolation depends also on the motion search precision. A half-pixel motion search approach for SI extraction is proposed in [KVO06].

Another interesting technique can be found in [MdQM07]. The authors propose to generate the SI using motion-based super-resolution approach, applied on the low resolution version of the frame $I_{wz}(x, y)$, which has been encoded by a conventional H.264-like codec in intra mode, and aided by other already decoded frames (at full resolution). We study a similar scenario in Section 3.4 of this work, describing the spatial side information.

Some works address the motion non-uniformity issue, and the problem of *blind* SI extractor: it must estimate the frame $I_{wz}(x, y)$ without knowing anything about it. Although the video content is usually varying rather slowly, the assumption of the motion uniformity does not always hold. This results in a temporal mismatch between the interpolated frame $I_{si}(x, y)$ and the original one $I_{wz}(x, y)$. The only possible solution to this problem is to transmit some additional data (usually called “hash” data) about the frame $I_{wz}(x, y)$, which would help synchronizing in time the two frames.

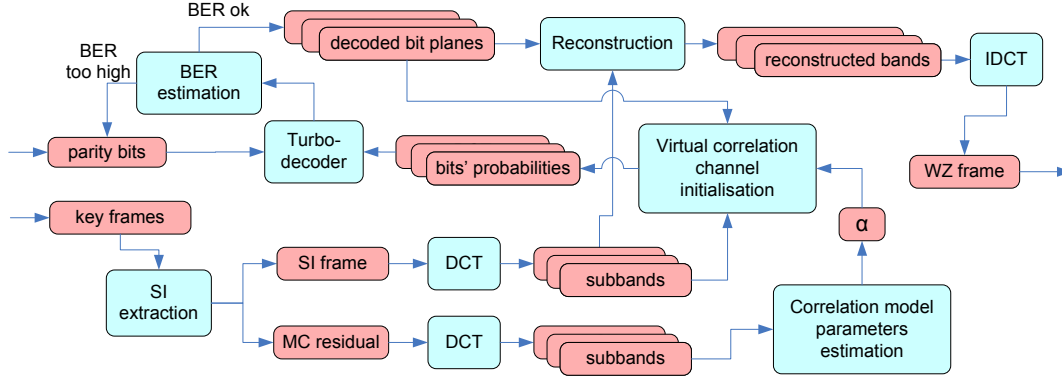


Figure 3.2: Detailed scheme of the WZ decoder

For example, some coarsely quantised DCT coefficients can be used as the hash information. This approach was proposed in [ARG04], and further developed in [AP07], introducing an adaptive choice of DCT coefficients to be transmitted as hash. Another type of hash information can be feature points information (e.g., their positions in the frame $I_{wz}(x, y)$, [MGM07]). However, the latter approach requires feature points tracking at the encoder thus increasing the encoder complexity, which is not always acceptable in distributed video coding. In fact, allowing intra frame processing at the encoder is making the architecture not fully distributed, or “quasi”-distributed, as the authors call it in the article.

The side information extraction and its relation to the final codec rate-distortion performance, has also been subject of several theoretic studies. So, Li and Delp prove in their work [LD05] that subpixel accuracy of the motion search is not of great importance for the SI extraction; on the other hand, decoding with multiple SI hypotheses (i.e. when SI extractor generates a set of side information instances, and not the only one) might be much more effective than with a single SI hypothesis. The work is further developed in [LLD07].

In this work, we make several contributions to the research on side information extraction. Section 3.1 describes a mesh-based motion compensated interpolation algorithm and its application to the SI extraction problem. Inspired by theoretical results obtained in [LD05], in Sections 3.2-3.3 we consider two different methods of using the two side information hypotheses in parallel, “Hybrid SI extraction”, and multihypotheses decoding. These sections are followed by our studies on usage of the spatial SI in DVC (Section 3.4), and on side information denoising (Section 3.5).

3.1 Mesh-based motion estimation and compensation

Block-based motion model is widely used in video processing and video compression because of its simplicity and cost efficiency. However, its capabilities of tracking only



Figure 3.3: 2D regular triangular mesh

translational motion are not always sufficient. For example, if the two frames are separated by a relatively long period of time, a complex motion (e.g. rotation, scaling) cannot be accurately approximated by translations of fixed-sized blocks. More complex motion models capable of describing affine transformations, are required. Mesh-based motion models represent a class of models which do have these capabilities. In this section we study the perspectives of applying a mesh-based motion model to the side information extraction in DVC.

The side information extraction process comprises three steps:

1. mesh construction in the base frame;
2. estimation of the mesh position in the reference frame;
3. bidirectional frame interpolation using estimated mesh-based motion model.

3.1.1 Mesh construction in the base frame

A Delaunay triangulation (see [dBvKOS00] for example) is used to construct the 2D mesh. Here a regular triangular mesh is imposed on the base frame to avoid requiring a segmentation mask, and from considerations of the decoder complexity. However, if such a segmentation mask would be available, the performance of the estimator could be improved by adapting the initially imposed triangulation to object borders. On Fig. 3.3 an example of a 2D regular triangular mesh is shown.

3.1.2 Mesh motion parameters estimation

The next step is estimating motion model parameters. A mesh-based motion model defines displacements of mesh nodes, i.e. their relative positions in the reference frame with respect to the base frame, while the motion trajectories for the remaining (not node) positions are obtained by interpolating the trajectories of the nodes.

The approach described below is based on works of F. Galpin [Gal02] and G. Marquant [Mar00]. If V is the set of all mesh nodes, then the model is defined by the set of

displacements $(dx_v, dy_v), v \in V$. Let C be a triangular cell of the mesh formed by three vertices $(v_1, v_2, v_3) \in V^3$, then the motion trajectory for any internal point $s \in C$ is given by:

$$\begin{cases} dx(s) &= w_1(s)dx_{v_1} + w_2(s)dx_{v_2} + w_3(s)dx_{v_3}, \\ dy(s) &= w_1(s)dy_{v_1} + w_2(s)dy_{v_2} + w_3(s)dy_{v_3}, \end{cases} \quad (3.3)$$

where $w_i(s), i = 1, 2, 3$, are barycentric coordinates of $s = (x, y)$ in the triangle C formed by vertices v_1, v_2 and v_3 , which verify by definition the equality $w_1(s) + w_2(s) + w_3(s) = 1$.

The goal of the motion estimation is to find a set of parameters $(dx_v, dy_v), v \in V$, which minimises the *displaced frame difference* (DFD) between the base frame I_{base} (at time instant t_1) and I_{ref} (at time instant t_2):

$$\{(dx_v^{opt}, dy_v^{opt})\}_{v \in V} = \arg \min_{\substack{dx_v, dy_v \\ v \in V}} \sum_{(x,y) \in \Omega} [I_{base}(x, y) - I_{ref}(x + dx(x, y), y + dy(x, y))]^2, \quad (3.4)$$

where Ω is the estimation support, which in our implementation is the whole frame I_{base} , and $dx(x, y), dy(x, y)$ are given by equation (3.3).

Applying Gauss-Newton sum-of-squares minimisation algorithm to the function $DFD(\{dx_v, dy_v\}_{v \in V}) = \sum_{(x,y) \in \Omega} [I_{base}(x, y) - I_{ref}(x + dx(x, y), y + dy(x, y))]^2$, the minimisation problem is turned into a problem of solving a system of $2|V|$ equations:

$$\begin{cases} \frac{\partial [DFD(\{dx_v, dy_v\}_{v \in V})]}{\partial dx_i} = 0, \\ \frac{\partial [DFD(\{dx_v, dy_v\}_{v \in V})]}{\partial dy_i} = 0, \\ i \in V. \end{cases} \quad (3.5)$$

Expanding the first equation in (3.5), and denoting $I(x, y) = I_{base}(x, y)$, $\hat{I}(x, y) = I_{ref}(x + dx(x, y), y + dy(x, y))$ for brevity, we obtain:

$$\begin{aligned} & \sum_{(x,y) \in \Omega} [I(x, y) - \hat{I}(x, y)] \frac{\partial}{\partial dx_i} [-\hat{I}(x, y)] = 0 \\ \Leftrightarrow & \sum_{(x,y) \in \Omega} [I(x, y) - \hat{I}(x, y)] \frac{\partial dx}{\partial dx_i} \frac{\partial}{\partial dx} \hat{I}(x, y) = 0 \\ \Leftrightarrow & \sum_{(x,y) \in \Omega} [I(x, y) - \hat{I}(x, y)] w_i(x, y) \nabla_x \hat{I}(x, y) = 0. \end{aligned} \quad (3.6)$$

Developing the term in square brackets for a small displacement increment $\delta \vec{u} = (\delta dx_m, \delta dy_m), m \in V$ (which permits to pass over from the previous step estimate $\hat{I}^k(x, y)$ to the next one $\hat{I}^{k+1}(x, y)$ in iterative Gauss-Newton method) we have:

$$\begin{aligned} I(x, y) - \hat{I}(x, y) &= I(x, y) - (\hat{I}^k(x, y) + \delta \hat{I}(x, y)) \\ &= I(x, y) - \hat{I}^k(x, y) - \sum_{m \in V} \left(\frac{\partial \hat{I}(x, y)}{\partial dx_m} \delta dx_m + \frac{\partial \hat{I}(x, y)}{\partial dy_m} \delta dy_m \right) \\ &= (I(x, y) - \hat{I}^k(x, y)) - \sum_{m \in V} \left(\frac{\partial \hat{I}(x, y)}{\partial dx} \frac{\partial dx}{\partial dx_m} \delta dx_m + \frac{\partial \hat{I}(x, y)}{\partial dy} \frac{\partial dy}{\partial dy_m} \delta dy_m \right) \\ &= -\nabla_x I(x, y) - \sum_{m \in V} w_m(x, y) (\nabla_x \hat{I}(x, y) \delta dx_m + \nabla_y \hat{I}(x, y) \delta dy_m). \end{aligned} \quad (3.7)$$

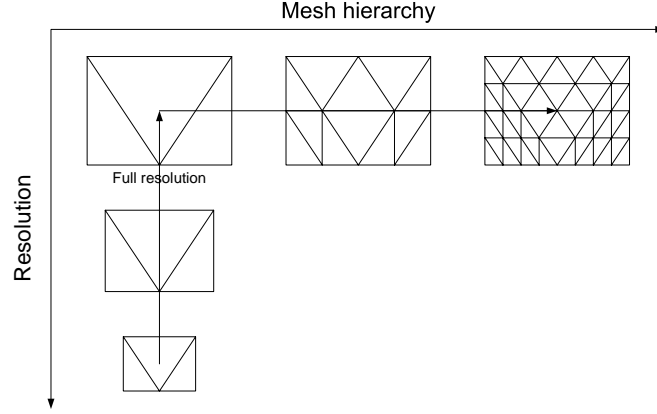


Figure 3.4: Hierarchical multi-resolution approach to minimisation

where $\nabla_t I(x, y)$ denotes the temporal gradient of the image along motion trajectories calculated with the current estimate of $\{(dx_v, dy_v)\}_{v \in V}$.

Performing similar development of the second equation in (3.5) and substituting (3.7) in (3.6), we finally obtain a system of $2|V|$ equations with $2|V|$ unknown variables $\{(\delta dx_m, \delta dy_m)\}_{m \in V}$:

$$\begin{cases} \sum_{(x,y) \in \Omega} w_i(x, y) \nabla_x \hat{I}(x, y) \left[\nabla_t I(x, y) + \sum_{m \in V} w_m(x, y) (\nabla_x \hat{I}(x, y) \delta dx_m + \nabla_y \hat{I}(x, y) \delta dy_m) \right] = 0, \\ \sum_{(x,y) \in \Omega} w_i(x, y) \nabla_y \hat{I}(x, y) \left[\nabla_t I(x, y) + \sum_{m \in V} w_m(x, y) (\nabla_x \hat{I}(x, y) \delta dx_m + \nabla_y \hat{I}(x, y) \delta dy_m) \right] = 0, \\ i \in V. \end{cases} \quad (3.8)$$

This sparse system (only few w_i and w_m are not zeros) is solved by a Levenberg-Marquard algorithm (see [Mar00] for more details).

To improve the robustness of the motion estimation, the minimisation is carried out using hierarchical multi-resolution approach (see Figure 3.4). Multiple resolution levels allow correct estimation of large displacements (for video sequences with high motion and long GOPs) without sacrificing the resolution of the estimation. In its turn, the hierarchy of meshes with different cell size improves the convergence speed and reduces complexity of the algorithm, preserving possibility of tracking small moving objects. Details about how the parameters are propagated from one hierarchy level to another, can be found in [Mar00].

3.1.3 Bidirectional motion compensated frame interpolation

The interpolation of the side information frame $I_{si}(x, y)$ at time instant t_{si} is done assuming motion uniformity during the time interval $[t_1, t_2]$ between the two frames $I_{ref}(x, y)$ and $I_{base}(x, y)$. Let k be the relative time shift, $k = \frac{t_{si} - t_1}{t_2 - t_1}$, and let $(v_x(t), v_y(t))$

denote coordinates of a mesh node $v \in V$ at the time instant t ; then the position $(m_x(t_{si}), m_y(t_{si}))$ of any mesh node $m \in V$ in the interpolated frame $I_{si}(x, y)$ is given by:

$$\begin{cases} m_x(t_{si}) &= (1 - k)m_x(t_1) + km_x(t_2), \\ m_y(t_{si}) &= (1 - k)m_y(t_1) + km_y(t_2), \end{cases} \quad (3.9)$$

Knowing mesh nodes coordinates at the time instant where the frame $I_{si}(x, y)$ has to be interpolated, the barycentric coordinates $w_i(x, y)$, $i = 1, 2, 3$ (in the respective mesh cell) can be computed for every pixel (x, y) of $I_{si}(x, y)$ with respect to the mesh triangle containing this pixel. After that, motion trajectories $(dx(x, y), dy(x, y))$ for each pixel (x, y) are interpolated using the barycentric coordinates from motion trajectories of respective mesh nodes as described in (3.3). Finally, the side information is obtained by interpolating every pixel $I_{si}(x, y)$ in the point on its motion trajectory, dividing this trajectory in proportion $k/(1 - k)$:

$$I_{si}(x, y) = kI_{base}(x - dx + kdx, y - dy + kdy) + (1 - k)I_{ref}(x + kdx, y + kdy). \quad (3.10)$$

3.1.4 Experimental results

The above approach to side information extraction using mesh-based motion model was evaluated in the DISCOVER codec, and the rate-distortion performance of the codec has been measured. In the first experiment, key frames quantisation is not taken into account; in other words, the key frames are compressed without losses. Although this is not a practical encoding scheme, it allows to better evaluate the precision and the spatial resolution of a motion compensated temporal interpolation (MCTI) method. The results for QCIF video sequences at 30Hz are shown on Fig. 3.5 (average rate and PSNR for all WZ frames are shown).

As one can see, the mesh-based approach provides significant improvement over the block-based one on sequences with low motion like “Hall Monitor” and “Coast-guard”. This is natural as the mesh-based motion model allows higher spatial resolution of the motion field than the block-based model does, which only specifies one motion vector per block, and thereby provides higher accuracy of interpolation.

On the other hand, it performs worse than the block-based approach on sequences with high motion, containing a lot of moving objects, occluding each other and the background (e.g. “Soccer”). This again is a usual behaviour of a mesh-based motion estimation which uses a single 2D mesh for representing the motion in the whole frame, and which is not capable of representing occlusions. Moreover, this type of video content violates the assumption of the motion continuity in the spatial domain. Indeed, the motion of each mesh node is optimised taking into account all six adjacent mesh triangles, or patches. However, some of these patches might belong to different objects, thus having different motion parameters; and the global optimisation in this case will not result in the best motion field for each of those objects. The problem

of motion discontinuity in 2D mesh-based motion estimation has been addressed in several works, see [Lau00], or [TTES96] for example. Nevertheless, in this work we do not address it directly. Instead, we propose a hybrid block/mesh-based solution described in Section 3.2.

In the second experiment key frames were compressed using a H.264-like encoder in intra mode. The quantisation parameter (QP) of the encoder was selected so that the average reconstruction quality of key frames was almost equal to that one of Wyner-Ziv frames. Test sequences has also been converted to the frame rate of 15Hz by discarding every even frame, as having QCIF video at 15Hz is more realistic than QCIF video at 30Hz. The results are presented on Fig. 3.6 (similarly to Fig. 3.5, the average rate and PSNR values are given for WZ frames only). In this experiment, the advantage the mesh-based SI extraction method is not certain, and varies for different sequences. It has been observed that in general case the advantage of the mesh-based approach over the block-based one is decreasing as the key frames quantisation parameter increases (providing higher compression rates for key frames), as shown on Fig. 3.7.

3.1.5 Conclusion on mesh-based motion estimation and compensation

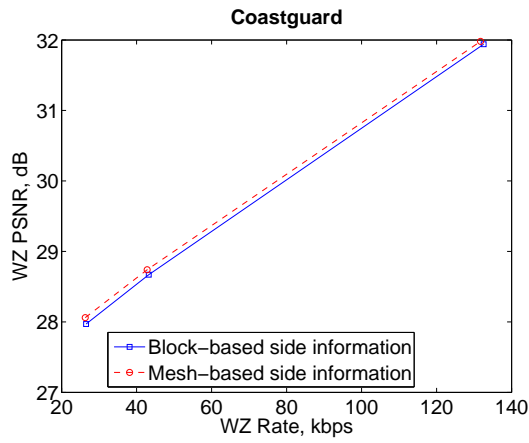
Analysing performances of the mesh-based motion estimation and compensation for side information extraction in DVC, we conclude that:

- the mesh-based motion model allows to obtain better side information than the block-based motion model (higher pixel-to-pixel correlation to WZ-frame) when the key frames distortion is low;
- the mesh-based motion estimation (ME) is less robust than the block-based ME to key frames quantisation, which is essential for encoding at low bit rates.

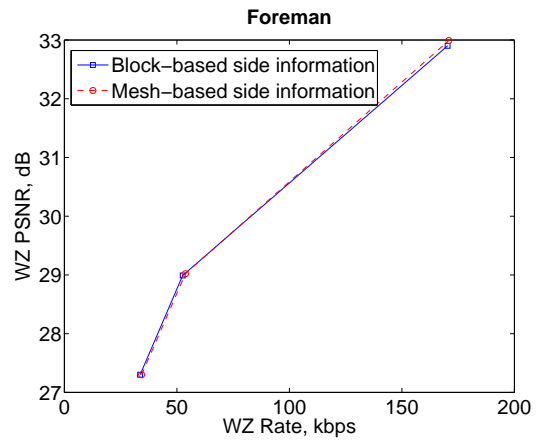
During the visual analysis of results given by both methods (block-based and mesh-based), it has also been noticed that it can be often difficult to prefer one method to the other, even on a single frame basis, i.e. some portions of the image are better interpolated using one method, while other portions are better with the other method. This observation gave start to a research direction described in Section 3.2, which consists in combining the two SI extraction approaches into a single method (called “Hybrid SI extraction”) in order to produce a SI of better quality than the two constituent methods (block-based and mesh-based) can provide separately.

3.2 Hybrid mesh-based/block-based interpolation

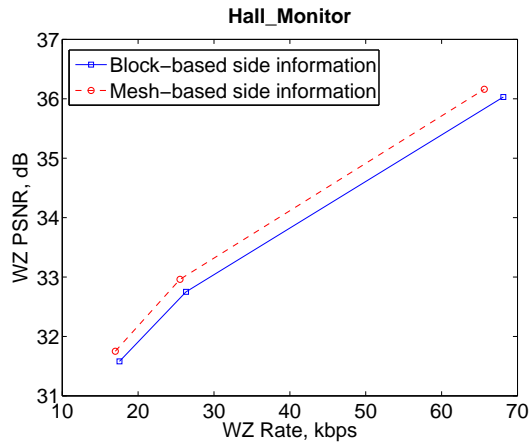
It was shown experimentally that the mesh-based motion model provides improved motion field accuracy compared to the block-based approach, especially when the



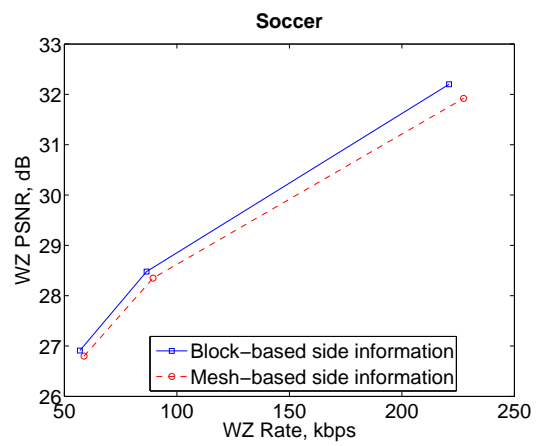
(a) Coastguard sequence



(b) Foreman sequence



(c) Hall Monitor sequence



(d) Soccer sequence

Figure 3.5: Rate-distortion performance of the WZ codec; key frames are compressed without losses

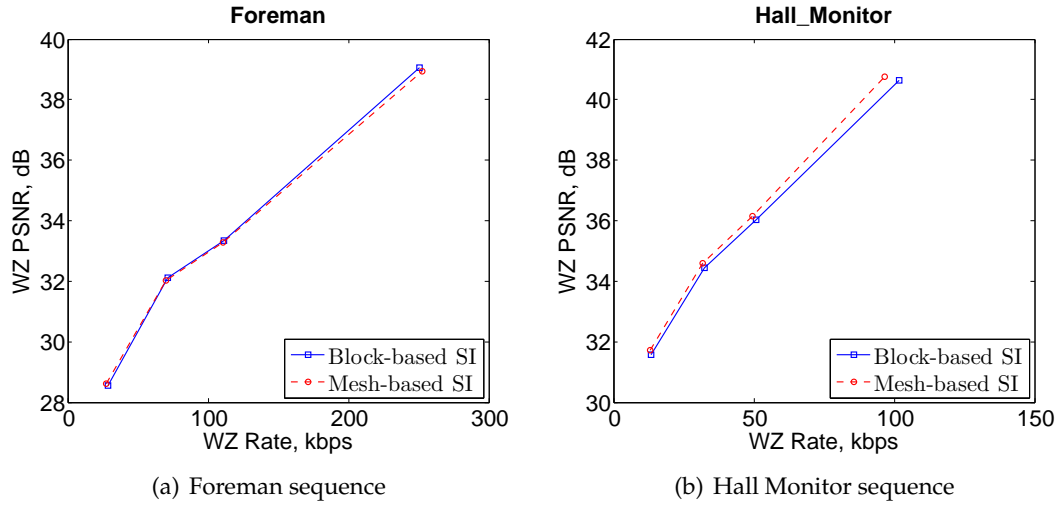


Figure 3.6: Rate-distortion performance of the WZ codec; key frames compressed to have approximately the same quality as WZ frames

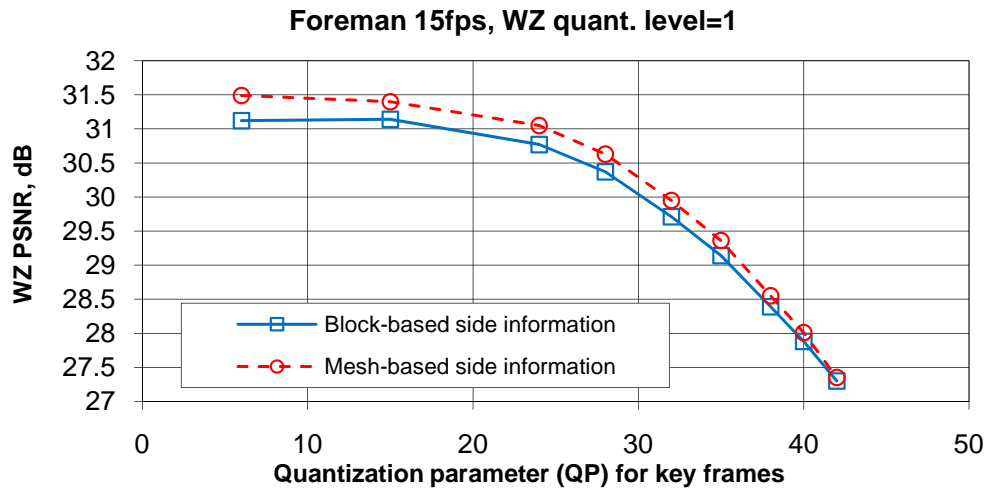


Figure 3.7: Side information quality as function of key frames quantisation parameter QP

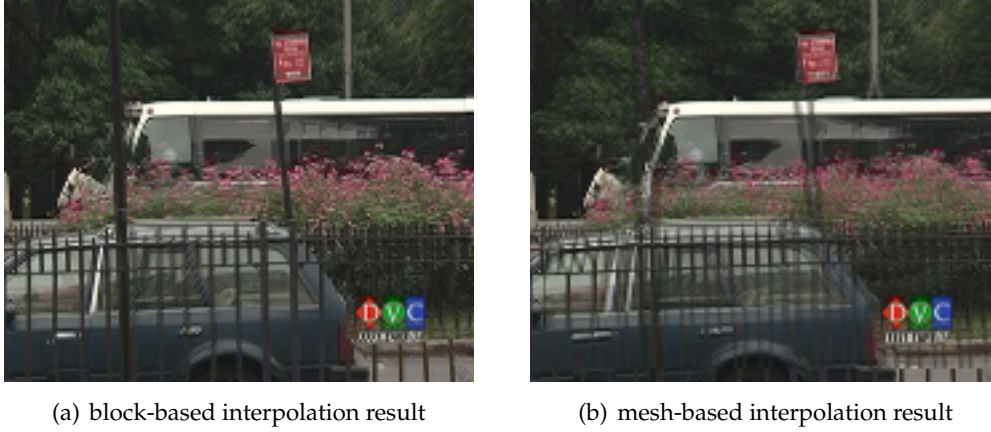


Figure 3.8: Problem with occlusions

motion is varying smoothly in the spatial domain. This includes motion caused by camera rotation, zooming, and probably translation, if the captured picture is affected by optical distortion (perspective).

However, if there are many objects with different motion characteristics in the frame, that occlude each other or the background, then the mesh-based approach becomes non effective, since it starts estimation with a regular mesh and therefore does not take into account object borders. On the other hand, block-based motion estimation can locally achieve better accuracy in these areas due to the local character of the estimation process (see Fig. 3.8 for example). This observation led to the development of a hybrid mesh-based/block-based frame interpolation technique which has been published in [KG06], and is described in this section.

Suppose that we have a measure of accuracy correlated with each SI extraction method, and that this measure can be used to determine which method performs locally better (on the part of the image) than the other, i.e. retains better correlation to the encoded WZ-frame. Then a hard (switching) or a soft (blending) decision based on this measure can be made independently for every image part, and the resulting SI frame must have higher correlation.

It's worth to note that designing such a measure might be quite challenging, as the decoder has no *a priori* information about the WZ-frame, and therefore cannot judge by the SI frame itself about its correlation to the WZ-frame. Thus unless extra information about the WZ-frame is sent additionally (e.g., "hash" from [AP07, ARG04], or "feature points" from [MGM07]), any measure will lack accuracy depending on the video sequence characteristics.

The case of sending extra information about WZ-frame is not considered here. Instead, we try to design measures that will do their best in absence of any supplementary information. We present our approaches for both pixel and transform domains.

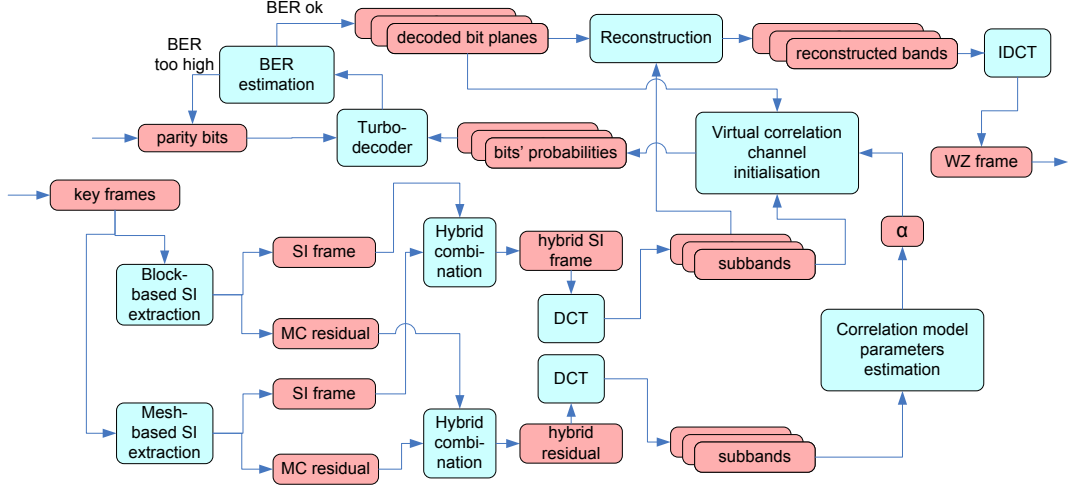


Figure 3.9: Scheme of the WZ decoder with hybrid SI extraction in pixel domain

We also propose a scheme for updating the measure during the decoding. Finally we discuss the results and perspectives of using hybrid interpolation for the DVC.

3.2.1 Hybrid interpolation in pixel domain

In spite of the fact that the DISCOVER architecture is only supposing transform-domain encoding of WZ frames, we start discussing the hybrid interpolation from the pixel domain case. The main reason for this is that it is simpler. Indeed, it is more natural to combine different SI extraction methods in the pixel domain as they do work in the pixel domain for obtaining the SI interpolated frame. The scheme of the decoder with hybrid SI construction in pixel domain is shown on Fig. 3.9.

Let $s = (s_x, s_y)$ represent a pixel position in a frame, and let $I_{base}(s)$, $I_{ref}(s)$, and $I_{si}(s)$ represent as before pixel luminance value at the position s in the base, reference and side information frames respectively. Let $\vec{u}^+(s)$ and $\vec{u}^-(s)$ be the forward and backward half motion vectors respectively, i.e. the motion model suggests that the following points are connected by motion trajectories:

$$I_{ref}\left(s + \vec{u}^-(s)\right) \leftrightarrow I_{si}(s) \leftrightarrow I_{base}\left(s + \vec{u}^+(s)\right) \quad (3.11)$$

Just to remind, the following is the interpolation rule for both block-based and mesh-based SI extraction methods:

$$I_{si}(s) = kI_{base}\left(s + \vec{u}^+(s)\right) + (1 - k)I_{ref}\left(s + \vec{u}^-(s)\right). \quad (3.12)$$

where $k = \frac{t_{si} - t_{ref}}{t_{base} - t_{ref}}$ is the coefficient depending on the temporal distance between the frames.

It is natural to suppose that the motion model is locally correct if the difference between the two interpolated pixels is small. Thereby the following function:

$$MSE(s, \vec{u}(s)) = \left[I_{base}(s + \vec{u}(s)) - I_{ref}(s + \vec{u}(s)) \right]^2 \quad (3.13)$$

can be used as the measure of the accuracy of a SI extraction method on a pixel basis. However, using this measure may result in a combination rule which is very unstable spatially because of the noise which is always present in video data. Instead, we can use the average of $MSE(s, \vec{u}(s))$:

$$MSE_{5 \times 5}(s, \vec{u}(s)) = \frac{1}{|W_2(s)|} \sum_{p \in W_2(s)} \left[I_{base}(p + \vec{u}(s)) - I_{ref}(p + \vec{u}(s)) \right]^2, \quad (3.14)$$

where $W_d(s)$ is a small rectangular window surrounding pixel s :

$$W_d(s) = \{p : (|p_x - s_x| \leq d \text{ and } |p_y - s_y| \leq d)\}. \quad (3.15)$$

A hard combination rule based on (3.14) can then be written as follows:

$$I_{hybr}(s) = \begin{cases} I_{si}^{block}(s) & \text{if } MSE_{5 \times 5}(s, \vec{u}_{block}(s)) \leq MSE_{5 \times 5}(s, \vec{u}_{mesh}(s)), \\ I_{si}^{mesh}(s) & \text{otherwise.} \end{cases} \quad (3.16)$$

The knowledge of the interpolated frame $I_{hybr}(s)$ is sufficient to the decoder if it knows the parameters of the correlation model. Otherwise, the residual frame resulting from the motion compensation of frames $I_{base}(s)$ and $I_{ref}(s)$, is used to determine the correlation model parameter as described in Section 2.3.4.6. Therefore, a hybrid residual frame corresponding to the hybrid side information, must be calculated.

We use the same rule for the residual combination as for the hybrid SI construction:

$$R_{hybr}(s) = \begin{cases} R_{si}^{block}(s) & \text{if } MSE_{5 \times 5}(s, \vec{u}_{block}(s)) \leq MSE_{5 \times 5}(s, \vec{u}_{mesh}(s)), \\ R_{si}^{mesh}(s) & \text{otherwise,} \end{cases} \quad (3.17)$$

where $R_{si}^{block}(s)$ ($R_{si}^{mesh}(s)$ respectively) is the motion compensated residual (2.61) computed with the motion field $\vec{u}_{block}(s)$ ($\vec{u}_{mesh}(s)$ respectively). After that, $R_{hybr}(s)$ is transformed, and correlation model parameters in the transform domain are given by (2.63).

An example of hybrid SI extraction is shown on Fig. 3.10. The individual results of block-based and mesh-based interpolations are shown on Fig. 3.10(a) and 3.10(b) respectively. Fig. 3.10(c) shows the binary mask corresponding to the switching rule used in (3.16), with grey coloured patches showing the image parts where the block-based interpolation was selected, while the black colour stands for areas where mesh-based interpolation was preferred. Finally, Fig. 3.10(d) presents the final result of the hybrid SI extraction method.

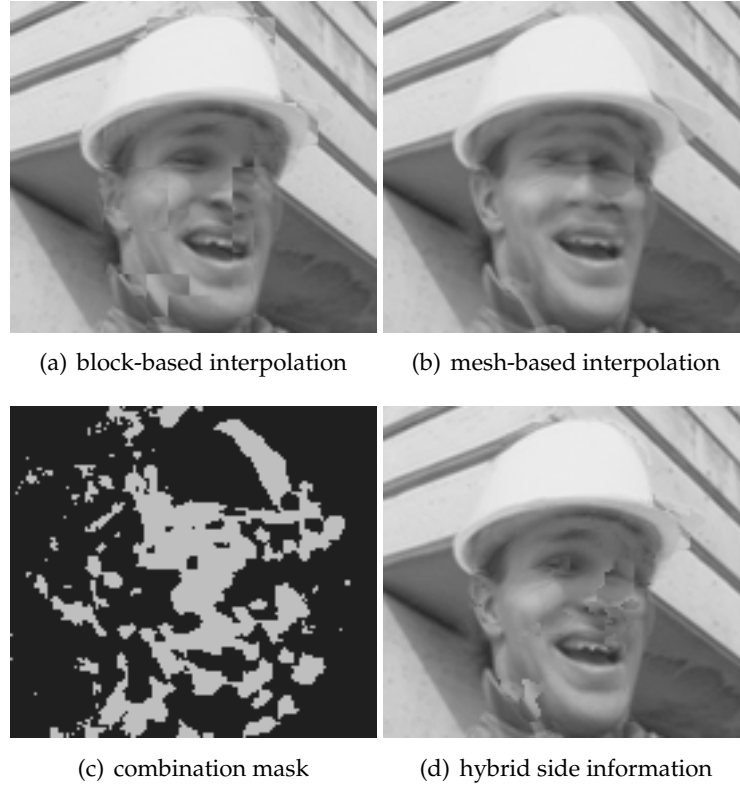


Figure 3.10: Example of hybrid SI extraction for Foreman sequence

3.2.2 Hybrid interpolation in transform domain

Now we pass to the description of the hybrid interpolation in the transform domain. That means that the side information is combined after the transform (integer number transform similar to DCT as described in Section 2.3.4.2, which we call below DCT for brevity), and not before as in the case of the hybrid interpolation in the pixel domain. The scheme of the decoder with hybrid SI construction in the transform domain is shown on Fig. 3.11.

The advantages of the hybrid interpolation in the transform domain are the following:

- the approach is more consistent with the correlation model which is defined in the transform domain (correlation between frequency bands);
- the combination can be made independently for each frequency band (possibly taking into account previously decoded frequency bands).

We are keeping here the same notation as in the previous subsection, denoting with $R(s)$ the motion compensated residual frame (2.61). We denote with $R^{DCT}(b, i, j)$ the

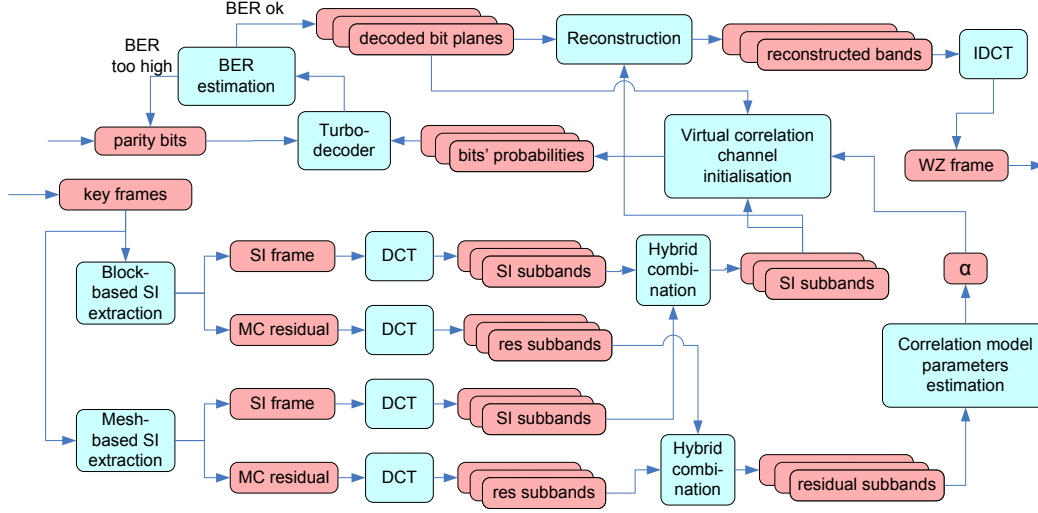


Figure 3.11: Scheme of the WZ decoder with hybrid SI extraction in transform domain

DCT coefficient of the band b ($b = 1, \dots, 16$ for a 4x4 DCT) at position (i, j) of the transformed image $R(s)$. The band $b = 1$ is the low frequency band, often called DC band (to distinguish it from other bands, which are called AC bands). Since DCT is an orthogonal transform, 16 coefficients $R^{DCT}(b, i_0, j_0), b = 1, \dots, 16$, at the same position (i_0, j_0) describe a block of 16 pixels of the non transformed image: $\{R(i_0 + l, j_0 + m)\}_{l,m=1,2,3,4}$.

The transition from the combination criterion in the pixel domain to the transform domain is possible due to the fact that DCT is a linear transform. That means if a 4x4 block in the residual image contains many non-zero pixels, the corresponding set of DCT coefficients after transformation will also have high energy. Given this observation, the absolute value of DCT coefficients of the residual can be used in the same manner as $MSE(s, \vec{u}(s))$ was used in the pixel domain:

$$I_{hybr}^{DCT}(b, i, j) = \begin{cases} I_{si_block}^{DCT}(b, i, j) & \text{if } |R_{block}^{DCT}(b, i, j)| \leq |R_{mesh}^{DCT}(b, i, j)|, \\ I_{si_mesh}^{DCT}(b, i, j) & \text{otherwise.} \end{cases} \quad (3.18)$$

The result of the hybrid interpolation $I_{hybr}^{DCT}(b, i, j)$ can be used directly by the transform domain decoder for the virtual correlation channel initialisation and for the final reconstruction. At the same time, if the correlation model parameters are not known, they can be estimated from the motion compensation residual $R_{hybr}^{DCT}(b, i, j)$ (see Section 2.3.4.6). Thus the transform domain residual $R_{hybr}^{DCT}(b, i, j)$ has to be also combined from the two transformed residuals, $R_{block}^{DCT}(b, i, j)$ and $R_{mesh}^{DCT}(b, i, j)$. We use the same rule for the residual combination, i.e.:

$$R_{hybr}^{DCT}(b, i, j) = \begin{cases} R_{block}^{DCT}(b, i, j) & \text{if } |R_{block}^{DCT}(b, i, j)| \leq |R_{mesh}^{DCT}(b, i, j)|, \\ R_{mesh}^{DCT}(b, i, j) & \text{otherwise.} \end{cases} \quad (3.19)$$

3.2.3 Improving hybrid interpolation with information of the decoder

In this section, we aim at improving the rule used in (3.19), which allows to switch between the block- and the mesh-based interpolation method. Our goal is to use the information of the decoder. Indeed, every decoded portion of data delivers an additional part of the information about the content of the WZ frame being decoded, thereby potentially reducing the handicap of the SI extractor, which has estimated the WZ frame without any information about it. In our architecture so far, we have only used this information to update the bit probabilities for the next bit plane (Section 2.3.2.2). However, the SI extraction itself can benefit from this extra information. A few approaches have been proposed in the literature.

The most common approach (e.g. presented in [ABP05b]) consists in refining the motion field with the extra available data before starting decoding of the next data portion. Although this technique works well in pixel domain, its application in transform domain is complicated. The return to the pixel domain is required for the motion field refinement, via the inverse DCT transform with only partially decoded coefficients. This results in a poor quality partial frame estimate, which is not usable for the refinement during first decoding steps.

Another solution called “motion learning”, which does not require the return to the pixel domain, was proposed in [VCFG08]. Although the side information is gradually improving during the iterative channel decoding, the successive decoding of bit planes and subbands is broken in the modified architecture, as all bit planes of all subbands are being decoded simultaneously. From the results presented in the paper, it is not clear if the sacrifice of inter-bit plane dependencies is recouped by the “motion learning” technique.

In this work we propose to exploit this extra information in a different way, namely to aid the hybrid SI extractor. We assume that if the motion estimation has failed locally, then for all subbands there will be a high probability of having a DCT coefficient value, which is far from the original value.

In other words, we propose to use previously decoded bands and bit planes to update the local measure of SI extraction accuracy, which we have initialised with the motion compensated residual in (3.18). As the update is done using the new information about the WZ frame, i.e. information that was not available earlier, an improvement of the SI can be expected. Experimental results justifying this assumption are shown in Section 3.2.5.

3.2.3.1 Combination rule using previously decoded bands

We present our approach in two parts, with the second part extending the first one. This is done mainly for the sake of simplicity. The first part describes SI updating taking into account only fully decoded bands, but not taking into account already

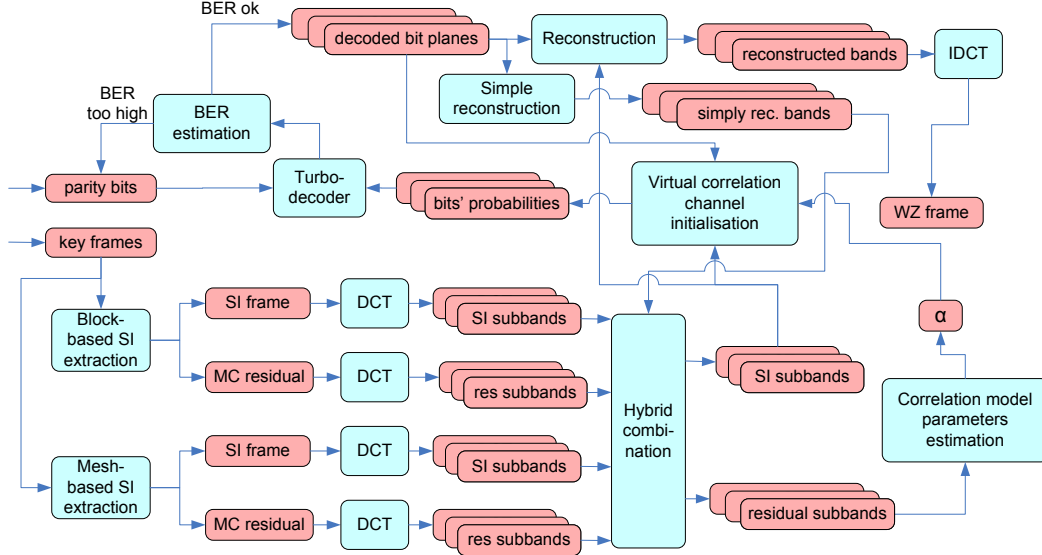


Figure 3.12: Scheme of the WZ decoder with hybrid SI extraction in transform domain updated during decoding

decoded bit planes of the same band. The corresponding WZ decoder scheme is presented on Figure 3.12. The difference with the previous scheme on Figure 3.11 is the additional feedback from the output of the turbo decoder to the hybrid combination block (which unites here both SI frame and MC residual hybrid combination blocks).

Please note that the simple reconstruction (on the feedback path) does not take the side information into account. The reason is that the side information value computed in (3.18), is biased with either block-based or mesh-based method, according to the hybrid combination rule's choice. Consequently, if the reconstruction procedure takes into account the SI, the reconstructed value will be also biased with the method selected by the hybrid combination rule. This should be avoided if the goal is to judge the correctness of this rule.

Let $q(b, i, j)$ be the quantisation index of the DCT coefficient $I_{wz}^{DCT}(b, i, j)$ of the WZ-frame $I_{wz}(x, y)$ being decoded. Note that $q(b, i, j)$ becomes available after decoding all bit planes of the band b . It designates a quantisation interval $[z_{i(q)}, z_{i(q)+1})$ for $I_{wz}^{DCT}(b, i, j)$ according to the adopted quantisation scheme (e.g. for the uniform quantization $\forall q : z_{i(q)+1} - z_{i(q)} = \Delta$, where Δ is the quantisation step). Hereinafter we omit q in the notation of the quantisation interval, and write simply $[z_i, z_{i+1})$ unless this short notation is ambiguous.

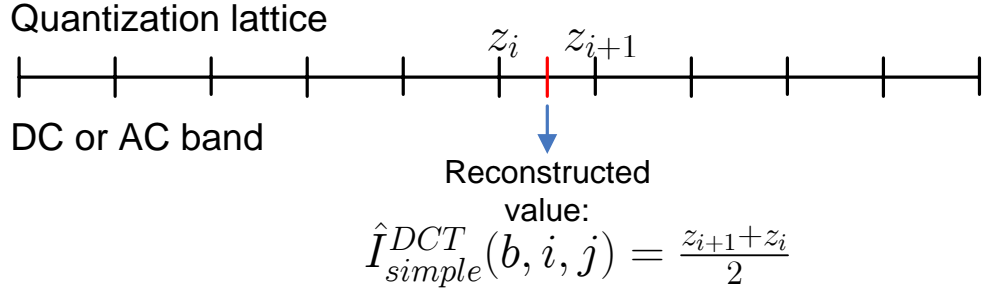


Figure 3.13: Simple reconstruction with uniform coefficient distribution

The simple reconstruction is given by the formula:

$$\hat{I}_{simple}^{DCT}(b, i, j) = E_{[z_{i+1}, z_i]} [I_{wz}^{DCT}(b, i, j)] = \frac{\int_{z_i}^{z_{i+1}} xp(x)dx}{\int_{z_i}^{z_{i+1}} p(x)dx}, \quad (3.20)$$

where $p(x)$ is the p.d.f. of $I_{wz}^{DCT}(b, i, j)$. Assuming that no information is available about $p(x)$, and supposing the uniform distribution over the quantisation interval $[z_i, z_{i+1})$ ($p(x) = p \forall x \in [z_{i+1}, z_i)$), (3.20) is simplified to:

$$\hat{I}_{simple}^{DCT}(b, i, j) = \frac{\int_{z_i}^{z_{i+1}} xp(x)dx}{\int_{z_i}^{z_{i+1}} p(x)dx} = \frac{p(z_{i+1}^2 - z_i^2)}{2p(z_{i+1} - z_i)} = \frac{z_{i+1} + z_i}{2}. \quad (3.21)$$

The simple reconstruction is illustrated in Fig. 3.13.

The hybrid combination rule for the band $b + 1$ is then modified as follows:

$$I_{hybr}^{DCT}(b + 1, i, j) = \begin{cases} I_{si_block}^{DCT}(b + 1, i, j) & \text{if } Err_{block}(b + 1, i, j) \leq Err_{mesh}(b + 1, i, j), \\ I_{si_mesh}^{DCT}(b + 1, i, j) & \text{otherwise.} \end{cases} \quad (3.22)$$

where

$$Err_{block}(b + 1, i, j) = |R_{block}^{DCT}(b + 1, i, j)| + \sum_{m=1}^b |\hat{I}_{simple}^{DCT}(m, i, j) - I_{si_block}^{DCT}(m, i, j)|, \quad (3.23)$$

$$Err_{mesh}(b + 1, i, j) = |R_{mesh}^{DCT}(b + 1, i, j)| + \sum_{m=1}^b |\hat{I}_{simple}^{DCT}(m, i, j) - I_{si_mesh}^{DCT}(m, i, j)|. \quad (3.24)$$

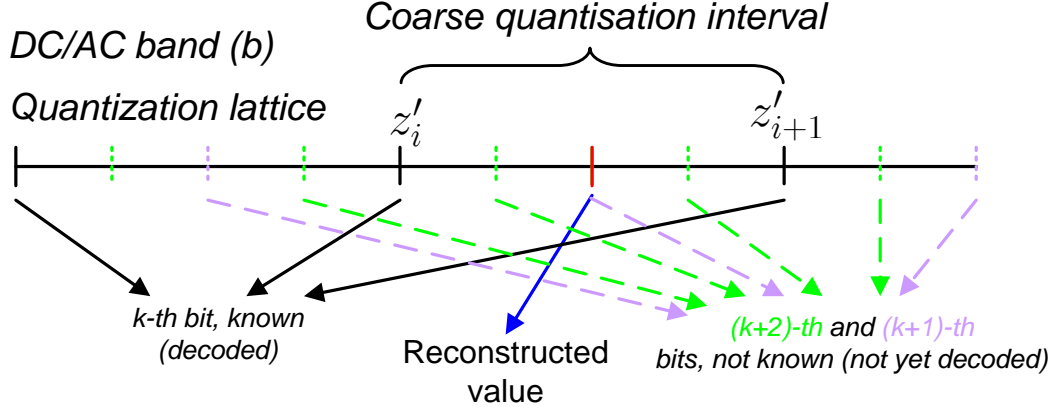


Figure 3.14: Simple intermediate reconstruction with uniform coefficient distribution

3.2.3.2 Combination rule using previously decoded bands and bit planes

Not only completely decoded subbands, but also each bit plane brings information about the subband it belongs to, and this information can be used to improve the hybrid combination. The scheme of the approach remains the same as depicted on Fig. 3.12. Only the frequency of the feedback is different: here it is exercised more often, i.e., after each successful bit plane decoding.

Suppose the band b is being decoded, that has M_b bit planes in total, and k of them are already successfully decoded, $0 < k < M_b$. The reconstruction of previous bands $m = 1, \dots, b-1$, (if any) is performed as in (3.20) or (3.21), as they have all their bit planes decoded. The intermediate reconstruction of the band b is more complicated and described below.

The fact that k of M_b bit planes are decoded means that up to k bits are known for each codeword $cw(b, i, j)$ representing the quantisation index $q(b, i, j)$. Depending on the length $len(b, i, j)$ of the codeword, $[k - len(b, i, j)]$ unknown bits can lead to the uncertainty about $q(b, i, j)$. This uncertainty means that at this stage the decoder considers $2^{k-len(b,i,j)}$ quantisation intervals $[z_i^1, z_{i+1}^1), [z_i^2, z_{i+1}^2), \dots$, for the coefficient $I_{wz}^{DCT}(b, i, j)$. The expectation of $I_{wz}^{DCT}(b, i, j)$ can then be computed similarly to (3.20):

$$\hat{I}_{coarse}^{DCT}(b, k, i, j) = E_{cw^k(b,i,j)} [I_{wz}^{DCT}(b, i, j)] = \frac{\sum_{w=1}^{2^{k-len(b,i,j)}} \int_{z_i^w}^{z_{i+1}^w} x p(x) dx}{\sum_{w=1}^{2^{k-len(b,i,j)}} \int_{z_i^w}^{z_{i+1}^w} p(x) dx}, \quad (3.25)$$

where $cw^k(b, i, j)$ represents the k first bits of the codeword $cw(b, i, j)$.

The expression can be readily computed under the uniform distribution assumption for $I_{wz}^{DCT}(b, i, j)$, if a fixed-length code (FLC) with lexicographic codeword assign-

ment is used to encode quantisation indices $q(b, i, j)$. Indeed, for a FLC with lexicographic codeword assignment the first k bits of the codeword $cw(b, i, j)$ designate a coarse quantisation interval $[z'_i, z'_{i+1}) = \bigcup_{w=1}^{2^{k-\text{len}(b,i,j)}} [z_i^w, z_{i+1}^w)$, which is *continuous* (see illustration on Fig. 3.14). Therefore, the same result as in (3.21) can be obtained:

$$\hat{I}_{coarse}^{DCT}(b, k, i, j) = \frac{\int_{z'_i}^{z'_{i+1}} xp(x)dx}{\int_{z'_i}^{z'_{i+1}} p(x)dx} = \frac{p((z'_{i+1})^2 - (z'_i)^2)}{2p(z'_{i+1} - z'_i)} = \frac{z'_{i+1} + z'_i}{2}. \quad (3.26)$$

The hybrid combination rule for the band b with k bit planes decoded is then modified as follows:

$$I_{hybr}^{DCT}(b, k, i, j) = \begin{cases} I_{si_block}^{DCT}(b, i, j) & \text{if } Err_{block}^{bp}(b, k, i, j) \leq Err_{mesh}^{bp}(b, k, i, j), \\ I_{si_mesh}^{DCT}(b, i, j) & \text{otherwise.} \end{cases} \quad (3.27)$$

where

$$Err_{block}^{bp}(b, k, i, j) = Err_{block}(b, i, j) + |\hat{I}_{coarse}^{DCT}(b, k, i, j) - I_{si_block}^{DCT}(b, i, j)|, \quad (3.28)$$

$$Err_{mesh}^{bp}(b, k, i, j) = Err_{mesh}(b, i, j) + |\hat{I}_{coarse}^{DCT}(b, k, i, j) - I_{si_mesh}^{DCT}(b, i, j)|. \quad (3.29)$$

3.2.3.3 Discussion on updating the hybrid interpolation

Here we sum up all the benefits and disadvantages of the proposed approach to updating the hybrid side information during the decoding process:

- + the decoder estimation of the WZ-frame is improved during the decoding, and used for decoding remaining bit planes and bands thus improving the overall rate-distortion performance of the codec (as confirmed by experimental results at the end of this section);
- + the complexity of the decoder can be reduced due to the reduced turbo decoder rate (and thus reduced number of turbo decoding iterations);
- the gain in the rate-distortion performance is not very big, as the most significant bands and bit planes are decoded first, and don't fully benefit from the SI quality improvement;
- the second part of the algorithm increases the decoder complexity as it causes certain operations (namely hybrid combination and correlation model parameters estimation) to be done multiple times for each band.

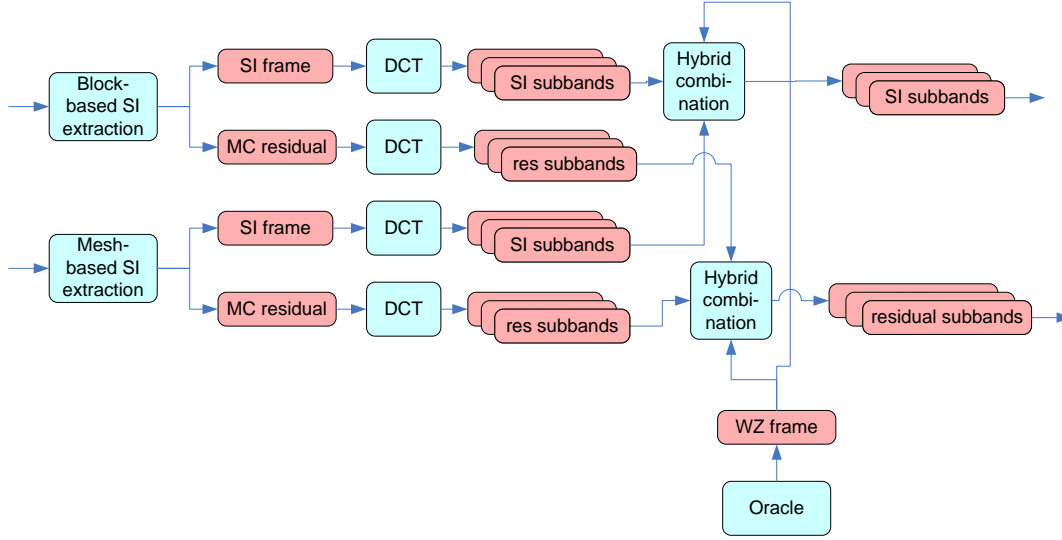


Figure 3.15: Hybrid SI extraction using oracle

3.2.4 Hybrid SI extraction using oracle

We were also interested in evaluating the potential of the hybrid SI extraction method, i.e. the upper bound of the performance that could be achieved if the hybrid combination rule would always select the correct SI value (closest to the encoded coefficient $I_{wz}^{DCT}(b, i, j)$). This can be easily simulated by granting the combination rule access to the value $I_{wz}^{DCT}(b, i, j)$ (Fig. 3.15). We call this approach “oracle-based hybrid SI extraction”. The oracle-based hybrid combination rule for the SI information and residual is the following (we omit the arguments (b, i, j)):

$$(I_{oracle}^{DCT}, R_{oracle}^{DCT}) = \begin{cases} (I_{si_block}^{DCT}, R_{block}^{DCT}) & \text{if } |I_{si_block}^{DCT} - I_{wz}^{DCT}| \leq |I_{si_mesh}^{DCT} - I_{wz}^{DCT}|, \\ (I_{si_mesh}^{DCT}, R_{mesh}^{DCT}) & \text{otherwise.} \end{cases} \quad (3.30)$$

3.2.5 Experimental results

The proposed Hybrid SI extraction algorithm was evaluated in the DISCOVER codec, and the rate-distortion performance of the codec has been measured. The experiment was conducted under realistic conditions, meaning that the key frames were also compressed (with encoding parameters permitting approximately the same reconstruction quality as for Wyner-Ziv frames). The proposed solution is compared to the reference block-based SI extraction solution (Section 2.3.4.5) on Fig. 3.16. The performance of the non-realistic oracle-aided Hybrid SI extraction is also plotted on graphs for comparison. Please note that unlike Figs. 3.5 and 3.6 showing rate and distortion for WZ frames only, here total average PSNR and rate values are plotted, i.e. including both key frames and WZ frames.

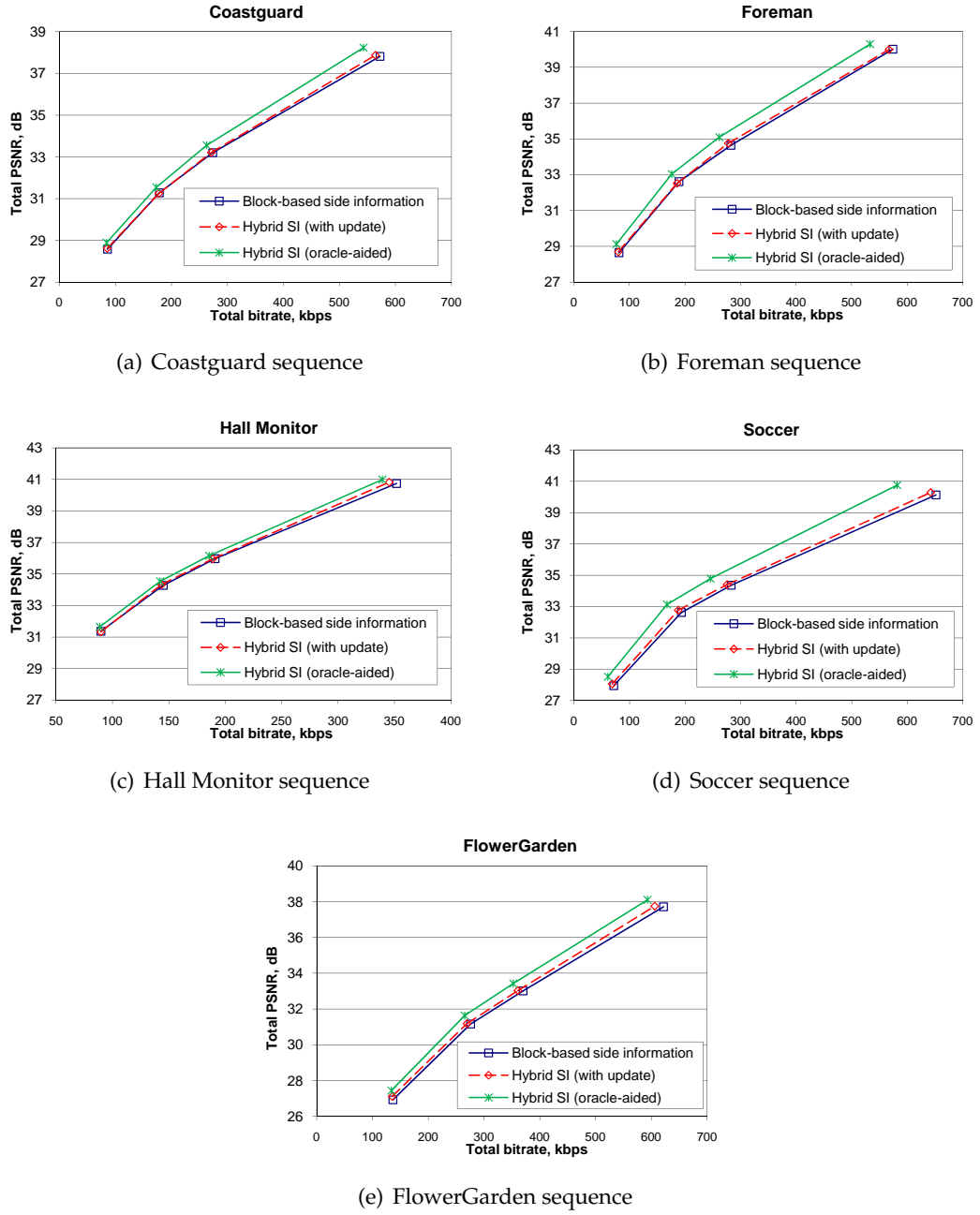


Figure 3.16: Rate-distortion performance of the WZ codec with Hybrid SI extraction

Plots show that the proposed method is competitive with the state-of-the-art block-based solution. It offers some improvement over the latter, although the gap between the curves is not very large, and varies for different sequences.

It is also clear that for certain sequences the oracle-aided solution, which uses the original frame, outperforms by far the realistic approach, having access to the motion compensated residual only. This fact suggests that the proposed combination scheme might be sub-optimal for those sequences, leaving a lot of space for further improvements. A possible improvement one can think of, is adding weighting coefficients into the update measure (3.28), e.g.:

$$Err^{bp}(b, k, i, j) = W_b |R^{DCT}(b, i, j)| + \sum_{m=1}^{b-1} W_m |\hat{I}_{simple}^{DCT}(m, i, j) - I_{si}^{DCT}(m, i, j)| + W_{b+1} |\hat{I}_{coarse}^{DCT}(b, k, i, j) - I_{si}^{DCT}(b, i, j)|, \quad (3.31)$$

However, in this work we did not focus on parameter tuning for particular sequences.

3.2.6 Conclusion on hybrid SI extraction

We have proposed an approach to improve the quality of the side information when multiple side information extractors are available. All formulae are written for two extractors, but they can be easily extended to any number of them.

Only hard combination ("switching") schemes have been considered. It is possible that better performance can be achieved with soft combination criteria ("blending") not covered by this study.

3.3 Multi-hypotheses decoding

The hybrid SI extraction described before, which is based on combination of side information issued by different SI extraction methods, is not the only way of taking into account multiple available side information. Another approach consists in decoding using multiple SI values *simultaneously*. Different SI values are called hypotheses, and the approach is called multi-hypotheses decoding. The method we present in this work is inspired by the work of Misra *et al* [MKR05]. It is schematically depicted on Fig. 3.17.

As it can be seen from the scheme, the WZ decoder needs to have its two blocks modified to accept two (or more) SI hypotheses, namely "Virtual correlation channel initialisation", and "Reconstruction". The necessary changes to these blocks are described below.

Another change is the use of multiple correlation models, one for each hypothesis. This is natural as different SI extractors may have different properties. Here we

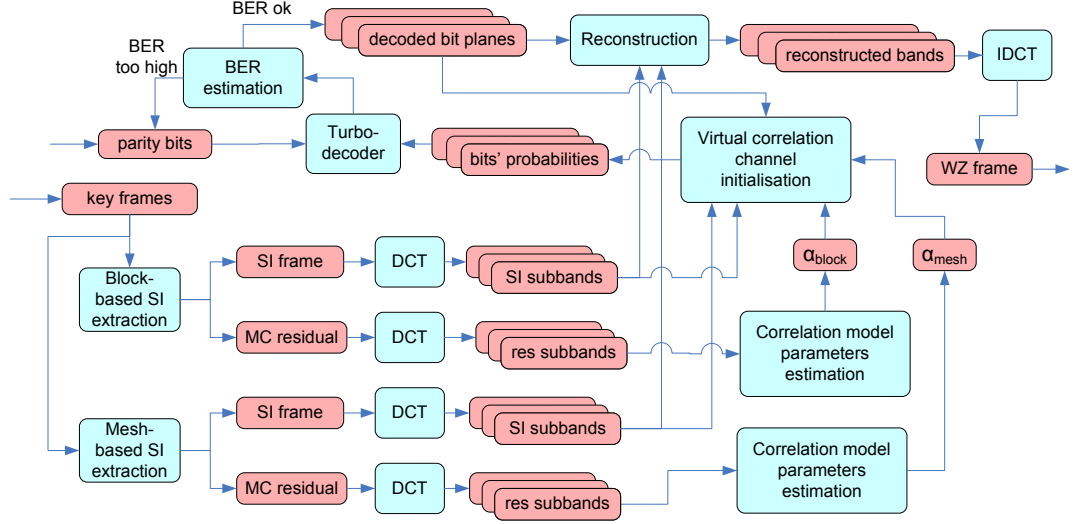


Figure 3.17: Multi-hypotheses WZ decoder scheme

assume that the type of the model is Laplacian for all SI extractors, thus only its parameters should be estimated. We adopt the algorithm described earlier in this work (see Section 2.3.4.6), for determining these parameters for each employed SI extraction method.

Before we start, let us introduce some notation which will shorten and simplify the following expounding. We denote with $X = I_{wz}^{DCT}(b, i, j)$ any band b of the original WZ frame (to be decoded), and with $Y_1 = I_{si_block}^{DCT}(b, i, j)$ and $Y_2 = I_{si_mesh}^{DCT}(b, i, j)$ the two corresponding SI hypotheses-bands. We also denote the bit planes of X with X^k , $1 \leq k \leq M_b$, where M_b is the number of bit planes in X . Small characters will refer to individual bits in the bit plane or DCT coefficients in the band, e.g. x, y_1, x^k .

3.3.1 Multi-hypotheses virtual correlation channel initialisation

Virtual correlation channel initialisation consists in estimating probabilities of each bit $x_{i,j}^k$ in position (i, j) to be 1 or 0, which is equivalent to receiving systematic bits of $x_{i,j}^k$ through a noise channel. Therefore the channel decoding of X^k can be performed using these probabilities and the parity bits (see Section 2.3.4 for more details).

These probabilities (called *a priori* probabilities to distinguish from *a posteriori* probabilities received at the issue of the turbo decoder) are calculated from the side information y (with imposed correlation model) and previously decoded bits of x :

$$\Pr(x^k | y, x^{k-1}, \dots, x^1). \quad (3.32)$$

Its calculation for the Laplacian correlation model was shown in Section 2.3.2.2.

When two SI hypotheses y_1 and y_2 are available (each with its own correlation model), they have to be taken into account in the calculation, meaning:

$$\Pr(x^k | y_1, y_2, x^{k-1}, \dots, x^1). \quad (3.33)$$

Please note that this probability is very difficult to express through probabilities $\Pr(x^k | y_1, x^{k-1}, \dots, x^1)$ and $\Pr(x^k | y_2, x^{k-1}, \dots, x^1)$ that can be easily computed, without knowing the model of correlation between Y_1 and Y_2 , and without knowing the distribution of X . One may try to define a complex (many dimensional) correlation model uniting X , Y_1 , and Y_2 , but there are several foreseen problems with this approach:

- the estimation of the model's parameters might be more computationally expensive;
- the calculation of $\Pr(x^k | y_1, y_2, x^{k-1}, \dots, x^1)$ itself might be more computationally expensive;
- the model will not be universal as it is a function of selected SI extraction methods; therefore it should be defined for every possible decoder configuration (with different SI extraction methods).

Instead, we propose a simple universal heuristic solution, consisting of taking the average of all method's probabilities:

$$\Pr(x^k | y_1, y_2, x^{k-1}, \dots, x^1) = \frac{\Pr(x^k | y_1, x^{k-1}, \dots, x^1) + \Pr(x^k | y_2, x^{k-1}, \dots, x^1)}{2}. \quad (3.34)$$

The advantage of this solution is that it can be easily computed for any number and any type of simple correlation models (the only requirement is to be able to calculate $\Pr(x^k | y_i, x^{k-1}, \dots, x^1)$ for each included SI extraction method).

3.3.2 Multi-hypotheses reconstruction

The task of the reconstruction module is to perform the inverse quantisation of DCT coefficients using the available side information. If $[z_i, z_{i+1})$ is the quantisation interval (which is known after turbo decoding of all bit planes $X^k, 1 \leq k \leq M_b$), then the minimum MSE reconstruction is given by:

$$\hat{x}_{\text{opt}} = E[x | x \in [z_i, z_{i+1}), y] = \frac{\int_{z_i}^{z_{i+1}} x f_{X|y}(x) dx}{\int_{z_i}^{z_{i+1}} f_{X|y}(x) dx}, \quad (3.35)$$

where $f_{X|y}(x)$ is the conditional p.d.f. of X given y . For the Laplacian model with parameter α we have:

$$f_{X|y}(x) = \frac{\alpha}{2} e^{-\alpha|x-y|}, \quad (3.36)$$

and the equation (3.35) has an equivalent closed-form expression, which will be derived later in Section 4.3.

When two SI hypotheses y_1 and y_2 are available (each with its own correlation model), they have to be taken into account in the calculation, meaning:

$$\hat{x}_{\text{opt}} = E [x|x \in [z_i, z_{i+1}), y_1, y_2] = \frac{\int_{z_i}^{z_{i+1}} x f_{X|y_1, y_2}(x) dx}{\int_{z_i}^{z_{i+1}} f_{X|y_1, y_2}(x) dx}. \quad (3.37)$$

From exactly same considerations as just before and similarly to (3.34), we adopt an heuristic approach to calculate $f_{X|y_1, y_2}(x)$, i.e.:

$$f_{X|y_1, y_2}(x) = \frac{f_{X|y_1}(x) + f_{X|y_2}(x)}{2}, \quad (3.38)$$

and the minimum MSE reconstruction is thus given by:

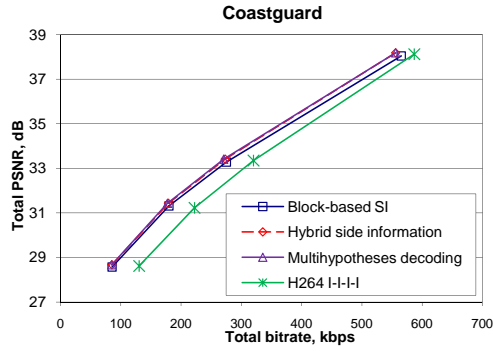
$$\hat{x}_{\text{opt}} = E [x|x \in [z_i, z_{i+1}), y_1, y_2] = \frac{\int_{z_i}^{z_{i+1}} x f_{X|y_1}(x) dx + \int_{z_i}^{z_{i+1}} x f_{X|y_2}(x) dx}{\int_{z_i}^{z_{i+1}} f_{X|y_1}(x) dx + \int_{z_i}^{z_{i+1}} f_{X|y_2}(x) dx}, \quad (3.39)$$

Noteworthy is the fact that the form (3.39) still permits easy calculation of \hat{x}_{opt} with the method, described in Section 4.3, thus making the adopted heuristic expression for $f_{X|y_1, y_2}(x)$ very attractive.

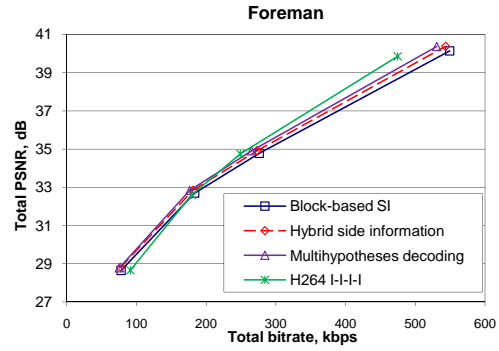
3.3.3 Experimental results

Multihypotheses decoding approach was evaluated in the adopted DVC architecture, and the rate-distortion performance of the codec has been measured. The experiment was conducted under realistic conditions, meaning that key frames were also compressed (with encoding parameters permitting approximately the same reconstruction quality as for Wyner-Ziv frames). On Fig. 3.18 it is compared to the reference block-based SI extraction solution described in Section 2.3.4.5, and to the Hybrid SI extraction method presented above in Section 3.2. As previously, total average PSNR and rate values are plotted, i.e. including both key frames and WZ frames. The performance of the H.264 codec in “I-I-I” mode (all frames of the sequence are encoded in intra mode) is also plotted on graphs for benchmarking purposes.

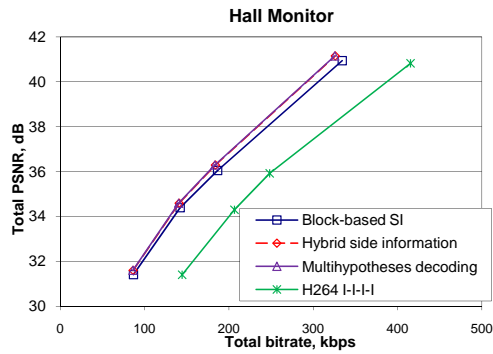
Plots show that in terms of rate-distortion performance multihypotheses decoding in general performs not worse than the Hybrid approach. For some sequences it allows approximately the same reconstruction quality level at smaller rate, but this improvement is not systematic.



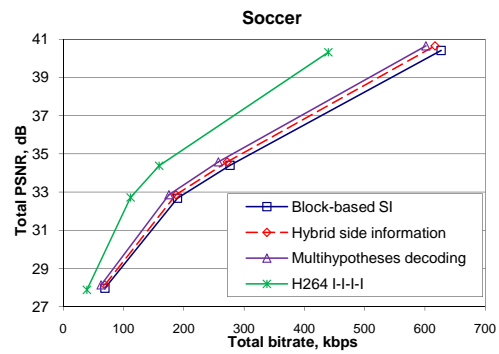
(a) Coastguard sequence, QCIF, 15Hz



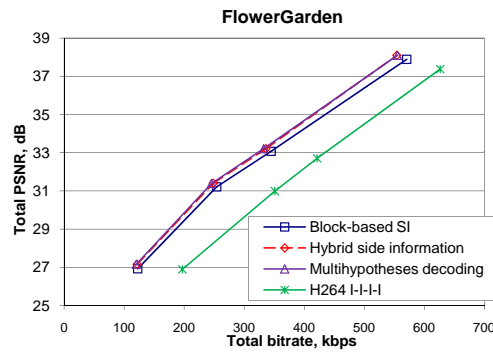
(b) Foreman sequence, QCIF, 15Hz



(c) Hall Monitor sequence, QCIF, 15Hz



(d) Soccer sequence, QCIF, 15Hz



(e) FlowerGarden sequence, QCIF, 15Hz

Figure 3.18: Rate-distortion performance of the WZ codec with Multihypotheses decoding in the transform domain

The comparison with the H.264 codec in intra mode is very important. It shows the current situation in distributed video coding, by comparing a state-of-the-art implementation of the classical video coding paradigm (which is based on prediction) to a state-of-the-art Wyner-Ziv video codec. Looking only at the RD-performance without touching the codec complexity question, which is crucial when comparing these two paradigms, it can be seen that the Wyner-Ziv scheme outperforms the intra mode of H.264 on video sequences containing low intensity motion (e.g. “Hall Monitor”), or containing mainly uniform global motion like camera panning (e.g. “Coastguard”, “FlowerGarden”). More detailed comparison of the two paradigms can be found in [dis07].

3.3.4 Conclusion on multi-hypotheses decoding

Multi-hypotheses decoding described in this section is an alternative to the Hybrid SI extraction approach, when multiple SI extractors are available. Its advantages over the Hybrid approach are lower decoder complexity and better robustness.

Indeed, in multi-hypotheses decoding approach no hard decision is taken, and thus all side information values (e.g., y_1 and y_2) contribute to the *a priori* bit probability $\Pr(x^k | y_1, y_2, x^{k-1}, \dots, x^1)$, balancing the influence of each other.

On the other hand, such a soft decision-approach might be unwanted if those values deviate from the true value x in a correlated manner, i.e. when $(y_1 - x)(y_2 - x) > 0$. In this case a good hard decision rule could perform better than taking into account both values, which correlated deviation might mislead the decoder.

The rate-distortion results shown on Fig. 3.18 confirm a marginal advantage of the multihypotheses approach over the Hybrid solution. Apparently, this advantage is due to the better robustness, as observed for sequences with complex motion where the constituent methods are not reliable. Coupled with lower decoder complexity, this tiny advantage makes the multihypotheses decoding scheme preferable to the other solution.

It is also worth to note that unlike the Hybrid SI extraction method, the described multihypotheses approach does not use sequential bit plane decoding. That means that its performance can be possibly improved by introducing weighting factors in (3.34) and (3.38).

3.4 Spatial side information

Until now we were only considering temporal side information, i.e. generated by temporal motion-compensated interpolation. We assumed that the correlation between adjacent motion-compensated video frames is high enough to permit effective WZ coding.

However, when the motion amplitude and/or character surpass the capabilities of the motion estimator, the temporal correlation might become low, thus making WZ coding with temporal SI ineffective. In such cases a usage of spatial side information may be considered.

A possible solution consists in switching between temporal and spatial SI depending on the correlation in both domains. For example, in [TTT⁺06a] and [AFAW06] authors propose to separate all pixels into two subsets and encode them separately, so that a decoder could choose between spatial and temporal SI while decoding the second subset. Unfortunately, both methods are designed for the pixel domain WZ architecture.

Straightforward application of these methods in transform domain (including partitioning into two subsets, and using one subset for the spatial SI generation for the other) will likely be ineffective. Indeed, the correlation between adjacent DCT coefficients is much lower than the correlation between samples, due to the increased distance between them (DCT coefficients represent blocks of 4x4 pixels).

Instead, we will concentrate on another approach. It supposes generation of a spatial SI by the encoder, and sending it along with parity bits to the decoder, when the usage of the spatial SI is preferable to the usage of the temporal SI (this can be determined either by simple motion detection at the encoder, or communicated by the decoder via the return channel). In our implementation, spatial SI consists of a subsampled version of the same frame, encoded with the key frame codec (H.264-type codec in "intra" mode). Varying the subsampling factor (from 2 to 4) and the quantisation parameter (QP) of the codec, different quality spatial SI information can be obtained.

The information we are sending as spatial SI, is very similar to "hash" data in [ARG04], however it is used in a different way. In [ARG04], it is used for aiding the motion estimation, thus for improving the temporal SI. In our turn, we are interested in evaluating the performance of the codec when this spatial side information replaces totally the temporal SI.

3.4.1 Spatial SI construction

The extraction of spatial SI has been implemented in Matlab. It comprises following steps (corresponding Matlab commands are given also):

Enc-1. Pre-filtering of the frame $I_{wz}(x, y)$ with Gaussian blur filter matrix H (of size 7x7, with $0.5 \leq \sigma \leq 1.0$):

```
H = fspecial('gaussian', 7, sigma);
frame = imfilter(frame, H, 'symmetric');
```

Enc-2. Downsampling of the filtered frame $H * I_{wz}(x, y)$ by a factor of two or four using bicubic resampling function:

```

resampler = makesampler('cubic', 'symmetric');
tform = maketform('affine', [1/factor 0 0; ...
                             0 1/factor 0; ...
                             0 0 1]);
frame = imtransform(frame, tform, resampler, ...
                    'Size', size(frame)/factor);

```

Enc-3. Encoding the downsampled version of the frame using H.264 encoder in intra mode:

```
system('h264encoder.exe -d h264.cfg');
```

Dec-1. Decoding the downsampled version of the frame using H.264 decoder (the reconstructed version of the video generated by the H.264 encoder is used).

Dec-2. Upsampling the decoded downsampled version to the full-sized frame:

```

frame = imtransform(frame, fliptform(tform), ...
                    resampler, 'Size', size(frame)*factor);

```

3.4.2 Performance analysis

The quality of the side information $I_{si}(x, y)$ is usually measured as peak signal-to-noise ratio (PSNR) with the original frame $I_{wz}(x, y)$, which is computed as follows:

$$PSNR(I_{wz}, I_{si}) = 20 \log_{10} \frac{255}{\sqrt{\frac{1}{N} \sum_{x,y} [I_{wz}(x, y) - I_{si}(x, y)]^2}}, \quad (3.40)$$

where N is the total number of pixels. The value is expressed in decibels (dB).

Experiments show that the quality of the temporal SI when the motion estimator has failed, is usually below 26 dB (see Fig. 3.19). Thus the parameters of spatial SI generation were adjusted accordingly, i.e. to have the quality of the spatial SI around 26 dB with the limitation that its rate was not higher than 1/3 of the compressed full-sized frame rate.

For example, on Fig. 3.20 both temporal and spatial SI frames are shown along with the original frame, and the same frame encoded with the key frame codec. The bit overhead caused by sending this spatial SI, is only 3720 bits, which is very low compared to the key frame bit rate for QIndex=8 (constituting about 31 kbits per key frame). At the same time, the increase in PSNR by more than 5 dB with respect to the temporal SI (and thus increase in correlation) could possibly make this bit investment totally compensated by the reduction of the parity bits rate, required for the WZ decoding.

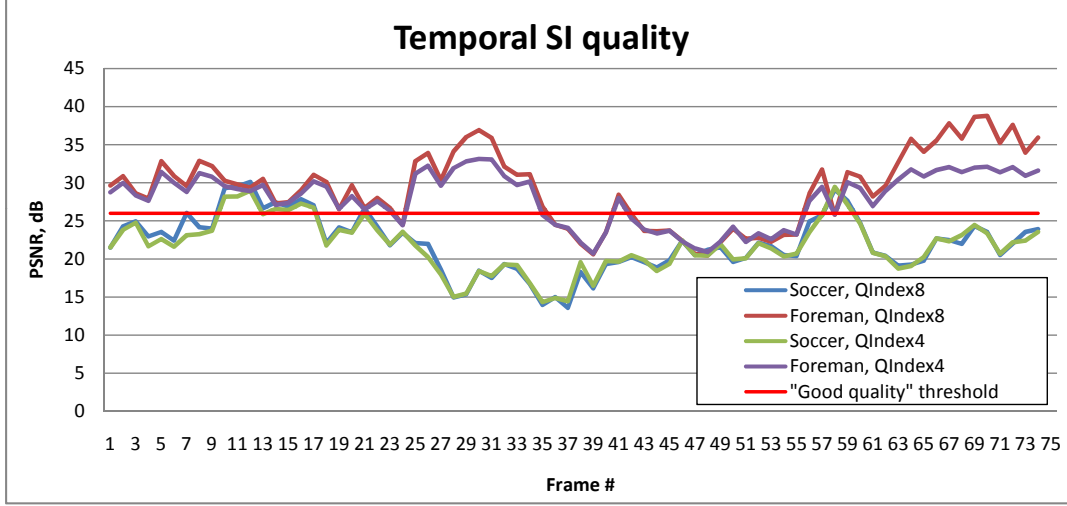


Figure 3.19: Evolution of the temporal SI in time for different sequences and compression options

Table 3.1: Decoding performances with spatial and temporal SI (Soccer, frame #2)

SI type	SI PSNR, dB	Parity bits rate, kbit	Decoding quality, dB
Temporal	22.04	39.75	38.05
Spatial	27.41	42.06	37.37

However, the WZ decoder demonstrates the opposite behaviour, as shown in Table 3.1: in spite of significantly increased SI PSNR, the parity bits rate is not decreased, but even increased (from 39.75 to 42.06 kbits), while the decoded frame quality is decreased at the same time.

The parity bits rate can be increased because of the two reasons:

1. the correlation model is not valid;
2. the correlation is reduced.

Thus, the first thing to verify is whether the Laplacian model still suits well the spatial SI, or a better model can be proposed. The following subsection describes this analysis.

3.4.3 Correlation model verification for spatial SI

We have studied the shape of the bandwise correlation histogram between the original frame $I_{wz}(x, y)$ and the spatial SI $I_{si_spat}(x, y)$, and compared it to the same histogram of the temporal SI $I_{si_temp}(x, y)$ (see Fig. 3.21). The correlation histogram, which is plotted on graphs, is the histogram of the difference $[I_{wz}^{DCT}(b, i, j) - I_{si_spat}^{DCT}(b, i, j)]$, where b is

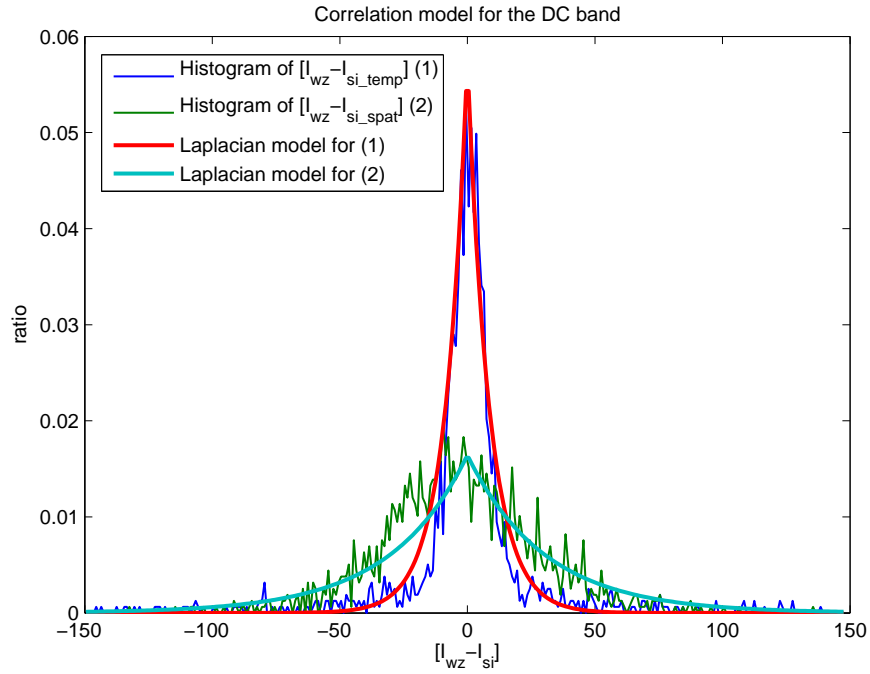


Figure 3.20: Temporal and spatial SI for the frame #2 of Soccer sequence

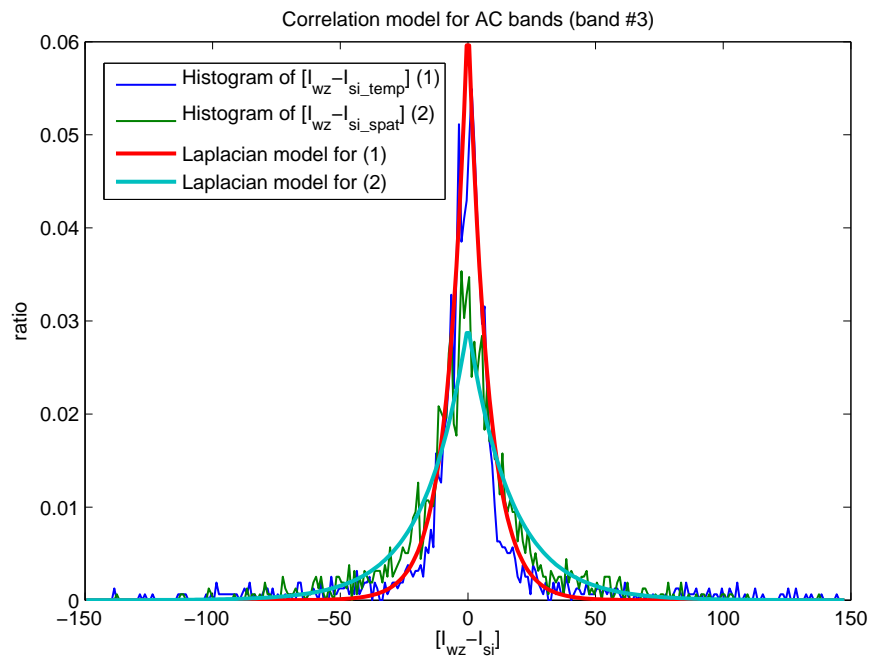
the band number. The histogram is plotted with curves instead of usual bars because using bars would overload the graph and make it unreadable.

As one can see from Figs. 3.21(a)-3.21(b), the shape of the correlation histogram can be well approximated by a Laplacian both for spatial and temporal SI. This is illustrated by superimposing a Laplacian distribution p.d.f. over the histograms. Please note that this superimposed Laplacian is not related to the actual correlation model that is used for decoding the corresponding band, as is only shown for illustration.

Thus, the Laplacian correlation model adopted for the temporal side information, is valid for the spatial SI as well. This means that the model itself was not the main reason of disagreement between the measure of quality of the SI (PSNR) and the decoding performance (seen in Table 3.1). Apparently, the correlation in DCT bands with the spatial SI is lower than the correlation with the temporal SI, but the PSNR value does not measure this correlation in an objective manner. This question is further investigated in the following subsection.



(a) DC band



(b) AC band

Figure 3.21: Correlation model analysis (DC and AC bands)

Table 3.2: $\sigma_{I_{si}}^2(b)$ per band b for spatial and temporal SI (Soccer, frame #2)

b (Band #)	$\sigma_{I_{si_spat}}^2(b)$	$\sigma_{I_{si_temp}}^2(b)$
1	928.14	10872.46
2	621.28	924.64
3	643.08	2230.17
4	574.07	634.73
5	335.83	497.18
6	614.23	351.85
7	149.40	84.79
8	202.27	169.40
9	193.75	209.61
10	135.55	152.29
11	53.67	64.27
12	107.23	96.06
13	56.40	46.68
14	31.31	27.56
15	27.40	27.66
16	12.27	12.58
Average	292.87	1025.18

3.4.4 A new quality measure for side information

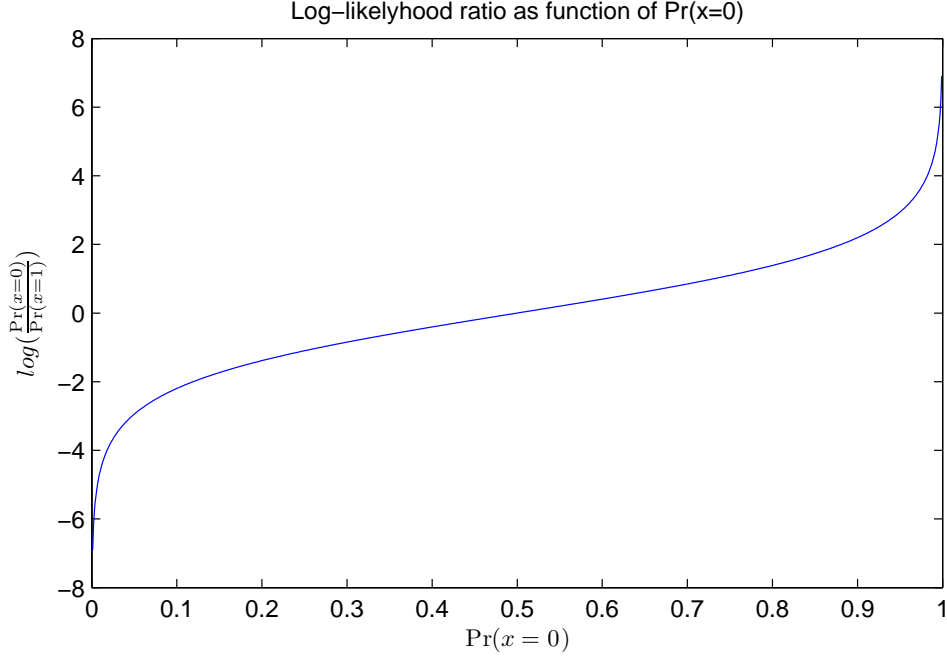
As described in Section 2.3.4.6, the Laplacian correlation model is assumed to be zero-centred (i.e., the expectation of error $[I_{wz}^{DCT}(b, i, j) - I_{si}^{DCT}(b, i, j)]$ is zero). The only parameter requiring estimation is the variance σ^2 of the distribution (although α is more often used, $\alpha^2 = \frac{2}{\sigma^2}$). The variance can be estimated as follows:

$$\hat{\sigma}_{I_{si}}^2(b) = MS E(I_{wz}^{DCT}(b, i, j), I_{si}^{DCT}(b, i, j)) = \frac{1}{D} \sum_{i,j} [I_{wz}^{DCT}(b, i, j) - I_{si}^{DCT}(b, i, j)]^2 \quad (3.41)$$

where $D = \frac{N}{16}$ is the number of DCT coefficients. This variance calculated for each DCT band of temporal and spatial SI, is shown in Table 3.2.

The average values in Table 3.2 are coherent with respective PSNR numbers. Moreover, values for separate bands show that the correlation between the spatial SI and the original frame is much higher for most DCT bands than the correlation between the temporal SI and the original frame. However, as was shown in Table 3.1, this advantage is not translated into the decoding performance advantage.

A study of this strange behaviour has evidenced that the reason of this disagreement is the square in MSE and PSNR formulae. In fact, this square comes from the assumption that the human eye is sensitive to the image reconstruction fidelity in a

Figure 3.22: Log-likelihood ratio as function of $\Pr(x = 0)$

quadratic manner. However, this might not be true when talking about decoder's sensitivity to errors in the side information.

Indeed, the side information is used to produce log-likelihood ratios (LLR) which are used by the channel decoder:

$$LLR = \frac{\Pr(x = 0)}{\Pr(x = 1)}. \quad (3.42)$$

Those LLRs are close to zero for $0.1 \leq \Pr(x = 0) \leq 0.9$ as can be seen on Fig. 3.22. This means that the sensitivity of the decoder to big errors in side information (which cause a quadratic function like MSE grow rapidly) is not much higher than its sensitivity to medium errors. In other words, in both cases the side information delivers very small information about the bit to be decoded. On the other hand, the LLR function grows rapidly near $\Pr(x = 0) = 0$ and $\Pr(x = 0) = 1$.

Therefore, we propose a new measure *SIQ* for evaluating the quality of the side information. It can be seen as a modification of PSNR, with the response closer to the form of the LLR function shown on Fig. 3.22. The idea is that this measure should better reflect the SI quality as it is required by the decoder:

$$SIQ(I_{wz}, I_{si}) = 20 \log_{10} \frac{255}{\frac{1}{N} \sum_{x,y} |I_{wz}(x, y) - I_{si}(x, y)|^{\frac{1}{2}}}. \quad (3.43)$$

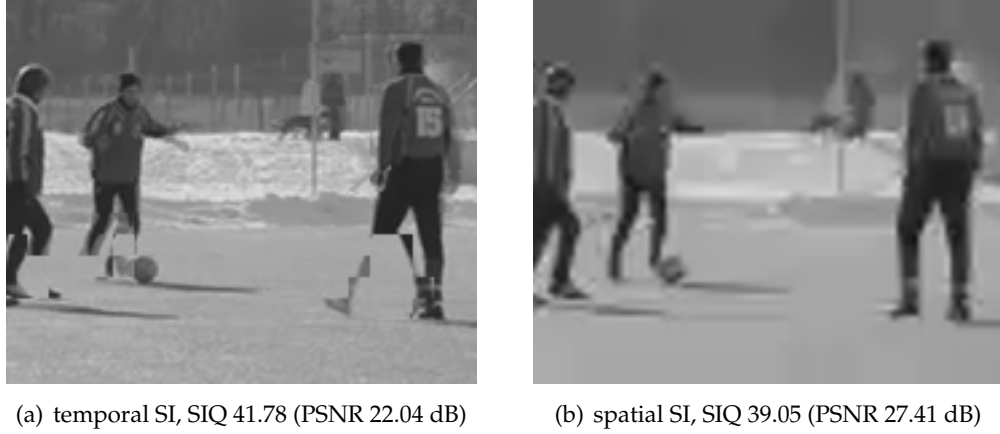


Figure 3.23: SIQ for temporal and spatial SI for the frame #2 of Soccer sequence

Fig. 3.23 shows the same images as Fig. 3.20, but with SIQ measure given instead of PSNR. The measure shows that the temporal SI outperforms the spatial method, which is consistent with results in Table 3.1.

3.4.5 Conclusion on spatial side information

The parameters of the spatial SI construction have been revisited to take into account the proposed SI quality measure. The downsampling factor 4 cannot be used anymore, as even the maximum quality obtainable with this factor (when the compression with the H.264 codec introduces no quality loss), does not allow to outperform the temporal SI.

With downsampling factor equal to 2, the decoding with the spatial SI starts to be more effective than the decoding with the temporal SI, when the quantisation parameter QP of the H.264 encoder goes down below 30-33. However, with this quantisation parameter the bit rate of the spatial SI is relatively high, and thus the total bit rate (including the bit rate of the spatial SI) remains always above the parity bits rate of the “only temporal SI”-decoder.

A conclusion can be drawn from this study concerning sending additional spatial side information along with parity bits. Due to the decoder’s high sensitivity to small errors in the side information (e.g., blurring), the loss of the spatial resolution (due to the downsampling/upsampling) has higher impact on the decoder than the fact that the number of big errors is reduced. Therefore, the usage of subsampled spatial SI *instead* of the temporal, is limited to situations where the motion estimation fails completely (e.g. if a scene change occurs).

Nevertheless, the fact that the spatial compression (i.e. compression using correlation in the spatial domain) is sometimes more effective than the temporal compression, is revisited further in this work, e.g. in Section 4.7, describing a mixed intra/WZ

encoding mode.

3.5 Side information denoising

Regarding the side information as a noisy version of the original frame, in this study we could not pass by the possibility of its improvement using denoising methods. This section briefly describes denoising techniques that have been tried, and presents some experimental results.

In this section for the sake of brevity we use the following short notation. Let $X = I_{wz}(x, y)$ be the original WZ-encoded frame, $Y = I_{si}(x, y)$ be the corresponding side information, and $Z = Y - X$ be the side information error, or the noise. A more convenient form of the latter equation for the decoder is $Y = X + Z$, as it only observes Y , and does not have access to X and Z .

The WZ decoder task is to estimate X using Y (and parity bits), thus to remove the noise Z . The theorem of Wyner and Ziv states that for a Gaussian noise Z , the parity bits rate, sufficient for reconstructing X with quadratic distortion not higher than D_X , depends on the noise variance σ_Z :

$$R(D_X) \geq R_{LB}(D_X, \sigma_Z) = \frac{1}{2} \log \frac{\sigma_Z^2}{D_X}. \quad (3.44)$$

Now, if Y is denoised, i.e. a new side information $Y' = X + Z'$ is obtained so that $\sigma_{Z'} < \sigma_Z$, then $R_{LB}(D_X, \sigma_{Z'}) < R_{LB}(D_X, \sigma_Z)$, and thus a reduction of parity bits rate is possible.

The general idea explained above with Gaussian noise, should be also valid for a Laplacian noise Z , although the lower bound for Wyner-Ziv coding with Laplacian correlation model is not established.

3.5.1 Denoising by wavelet thresholding

The bases of the wavelet theory and wavelet thresholding are not covered here, and the reader can refer to the book of Ingrid Daubechies [Dau92], for example, to have an introduction into the subject, as well as theoretical foundations of the wavelet theory.

Wavelet thresholding is a widely used technique for image denoising (see [CYV00] for example). Two methods are most frequently used: soft thresholding (also called coefficient shrinkage) and hard thresholding. The former consists in shifting all wavelet coefficients $w(u, v)$ towards zero by a certain value T_s :

$$\hat{w}_s(u, v) = \begin{cases} w(u, v) - T_s, & \text{if } w(u, v) > T_s, \\ w(u, v) + T_s, & \text{if } w(u, v) < -T_s, \\ 0, & \text{otherwise.} \end{cases} \quad (3.45)$$

The hard thresholding method consists in setting to zero all wavelet coefficients which absolute value is below T_h :

$$\hat{w}_h(u, v) = \begin{cases} 0, & \text{if } |w(u, v)| \leq T_h, \\ w(u, v), & \text{otherwise.} \end{cases} \quad (3.46)$$

Please note that thresholding is applied only to wavelet coefficients, while the low resolution coefficients are left untouched. Once the thresholding rule (soft or hard) is selected, the performance of the method depends on the applied wavelet transform and the choice of the threshold (T_s or T_h respectively).

In this study we use the denoising approach proposed in [CG06], with a few changes. The authors of [CG06] develop a new wavelet transform which they call the oriented wavelet transform (OWT), and demonstrate that it is superior when used for denoising, to several other wavelet transforms including separable wavelets, contourlets and bandelets. The oriented wavelet transform itself is described in more details in Section 5.1.2, where it is proposed as a replacement for the DCT in transform-domain Wyner-Ziv coding. Thus here we only cover the denoising algorithm.

In [CG06] it is proposed to use the hard thresholding method (3.46) with the same threshold $T = 3\sigma_Z$ for all subbands, where σ_Z is the variance of the noise in the pixel domain (supposed to be known or estimated by the denoiser). Our change to the algorithm concerns the way the orientation map is obtained, which is needed for wavelet decomposition. In the referenced work the true orientation map (corresponding to the noiseless image) is assumed unknown, thus it is modeled as a Markov field, and estimated before decoding. In our experiments below, we assume it to be known perfectly by the denoiser (e.g. calculated and transmitted by the encoder).

Several frames of two sequences Soccer and Foreman (QCIF) were selected to test the denoising approach (shown on Fig. 3.24). The first experiment consisted in evaluating the performance with the synthetic Laplacian noise added to these pictures. Fig. 3.25 shows two examples of noisy images, while Fig. 3.26 shows the same images denoised with the described algorithm. Results of the first experiment are summarised in Tables 3.3-3.4. They demonstrate that the algorithm is also efficient for a synthetic Laplacian noise removal (similar gains are obtained as reported in [CG06] for a synthetic Gaussian noise).

The second experiment was carried out on real data, i.e. to “denoise” the side information. The side information was generated using block-based MCTI as described in Section 2.3.4.5. Similarly to the first experiment, two noise levels were considered, corresponding to different QIndex encoder parameter values, 4 and 8 (which implies different compression settings for key frames also, thus different quality of MCTI result). Example frames with respective $\hat{\sigma}_N$ values (estimated using the original frame) are shown on Fig. 3.27. All results of the experiment are summarised in Tables 3.5-3.6. Unfortunately, the algorithm turned out to be far less efficient on real data (side information) than on synthetic noise. Moreover, in spite of the fact that the PSNR is



(a) Foreman, frame #28



(b) Soccer, frame #2



(c) Foreman, frame #86



(d) Soccer, frame #56



(e) Foreman, frame #108



(f) Soccer, frame #78

Figure 3.24: Frames selected for denoising experiments



(a) Foreman(#86) with Laplacian noise $\sigma = 15$



(b) Soccer(#2) with Laplacian noise $\sigma = 25$

Figure 3.25: Frames with synthetic Laplacian noise added



(a) Denoised, foreman(#86) ($\sigma = 15$)



(b) Denoised, soccer(#2) ($\sigma = 25$)

Figure 3.26: Denoised frames (synthetic Laplacian noise)

Table 3.3: Wavelet thresholding with synthetic Laplacian noise, Foreman

Sequence	Foreman					
frame #	28		86		108	
σ_N	15	25	15	25	15	25
Noisy, PSNR (dB)	24.55	20.13	24.65	20.17	24.60	20.18
Denoised, PSNR (dB)	30.34	27.75	30.50	27.91	30.21	27.88
Noisy, SIQ	38.04	35.84	38.09	35.86	38.07	35.88
Denoised, SIQ	41.56	40.11	41.74	40.24	41.46	40.14

Table 3.4: Wavelet thresholding with synthetic Laplacian noise, Soccer

Sequence	Soccer					
frame #	2		56		78	
σ_N	15	25	15	25	15	25
Noisy, PSNR (dB)	24.62	20.18	24.63	20.12	24.60	20.19
Denoised, PSNR (dB)	31.13	28.21	32.87	29.90	31.02	28.45
Noisy, SIQ	38.07	35.87	38.09	35.82	38.07	35.85
Denoised, SIQ	41.63	40.17	42.43	40.95	41.92	40.40

sometimes improved by up to 0.5 dB, the SIQ measure is always decreased, indicating that this improvement is not very likely to be preserved in the overall codec performance.

Some overall rate-distortion results are presented in Table 3.7. It shows a marginal improvement of the bit rate (the bit rate is lower when the SI is denoised), while the reconstruction PSNR is slightly decreased. Given that the conditions for the denoising algorithm were propitious (perfect knowledge of the noise variance, and of the true orientation map), we conclude on this subject that this denoising approach (denoising by wavelet thresholding) is not applicable in DVC, unless considerable changes are made in order to better accommodate the algorithm to the noise model of the side information.

3.5.2 Denoising using adaptive support transforms

We have also tried the denoising approach called BM3D (for “Block-matching and 3D filtering”) [DFKE07]. It is one of the two denoising approaches (the second is called SA-DCT for “Shape-adaptive DCT” [FKE07]) developed in the Signal Processing department of Tampere University of Technology. The general idea both approaches reside on, is to find a good support for a sparse transform (DCT for example) so that the separation between the signal and the noise was done by this transform in the most efficient way.

The first step of the BM3D method is to form three-dimensional data arrays called

(a) SI foreman(#86), QIndex=4, $\hat{\sigma}_N = 16.36$ (b) SI foreman(#86), QIndex=8, $\hat{\sigma}_N = 16.68$ (c) SI soccer(#2), QIndex=4, $\hat{\sigma}_N = 21.38$ (d) SI soccer(#2), QIndex=8, $\hat{\sigma}_N = 21.46$

Figure 3.27: Example frames of side information

Table 3.5: Wavelet thresholding with real data, Foreman

Sequence	Foreman					
frame #	28		86		108	
QIndex	4	8	4	8	4	8
$\hat{\sigma}_N$	11.27	11.00	16.36	16.68	16.52	17.73
Side info, PSNR (dB)	27.09	27.30	23.86	23.69	23.77	23.16
SI denoised, PSNR (dB)	26.87	27.10	24.17	24.13	24.25	23.70
Side info, SIQ	41.48	42.50	40.03	40.35	40.11	40.39
SI denoised, SIQ	40.99	41.54	39.55	39.71	39.73	39.84

Table 3.6: Wavelet thresholding with real data, Soccer

Sequence	Soccer					
frame #	2		56		78	
QIndex	4	8	4	8	4	8
$\hat{\sigma}_N$	21.38	21.46	45.24	45.57	38.19	39.76
Side info, PSNR (dB)	21.53	21.50	15.02	14.96	16.49	16.14
SI denoised, PSNR (dB)	21.73	21.68	15.34	15.30	16.88	16.68
Side info, SIQ	40.21	41.62	35.69	35.51	37.09	36.99
SI denoised, SIQ	39.72	40.03	35.51	35.35	36.79	36.79

Table 3.7: Rate-distortion performances on several frames (with and w/o wavelet thresholding)

Sequence	Foreman				Soccer			
Frame #/QIndex	86/QIndex=8		108/QIndex=8		56/QIndex=4		78/QIndex=8	
Rate/Distortion	R, kbit	D, dB	R, kbit	D, dB	R, kbit	D, dB	R, kbit	D, dB
SI not denoised	47.74	37.75	19.67	31.31	29.51	31.53	59.29	37.77
SI denoised	46.88	37.23	19.47	31.37	29.37	31.36	57.64	37.00

Table 3.8: BM3D algorithm test with real data, Foreman

Sequence	Foreman					
frame #	28		86		108	
QIndex	4	8	4	8	4	8
$\hat{\sigma}_N$	11.27	11.00	16.36	16.68	16.52	17.73
Side info, PSNR (dB)	27.09	27.30	23.86	23.69	23.77	23.16
SI denoised, PSNR (dB)	27.17	27.40	24.16	24.11	24.02	23.51
Side info, SIQ	41.48	42.50	40.03	40.35	40.11	40.39
SI denoised, SIQ	41.32	41.94	39.89	40.10	39.93	40.18

Table 3.9: BM3D algorithm test with real data, Soccer

Sequence	Soccer					
frame #	2		56		78	
QIndex	4	8	4	8	4	8
$\hat{\sigma}_N$	21.38	21.46	45.24	45.57	38.19	39.76
Side info, PSNR (dB)	21.53	21.50	15.02	14.96	16.49	16.14
SI denoised, PSNR (dB)	21.63	21.54	15.25	15.21	16.85	16.55
Side info, SIQ	40.21	41.62	35.69	35.51	37.09	36.99
SI denoised, SIQ	39.87	40.25	35.75	35.61	37.20	37.21

groups. These arrays are formed by stacking together all two-dimensional blocks of pixels that are similar in terms of the selected distance measure, e.g. mean-squared error (MSE), to the current block. Then, these groups are transformed using 3D DCT. The denoising is then achieved by shrinking transform coefficients, similarly to the method described in Section 3.5.1. In our SI denoising experiments we used a Matlab implementation of the algorithm provided on the web site [Foi].

Denoising results for the algorithm BM3D are presented in Tables 3.8-3.9, while the overall rate-distortion performance for the selected frames is given in Table 3.10. We note that in spite of lower PSNR values for denoised images, the overall rate-distortion performance is better with this BM3D method than with the presented wavelet thresh-

Table 3.10: Rate-distortion performances on several frames (with and w/o BM3D algorithm)

Sequence	Foreman				Soccer			
Frame #/QIndex	86/QIndex=8		108/QIndex=8		56/QIndex=4		78/QIndex=8	
Rate/Distortion	R, kbit	D, dB	R, kbit	D, dB	R, kbit	D, dB	R, kbit	D, dB
SI not denoised	47.74	37.75	19.67	31.31	29.51	31.53	59.29	37.77
SI denoised	46.68	38.29	19.67	31.53	28.91	32.15	56.18	38.51

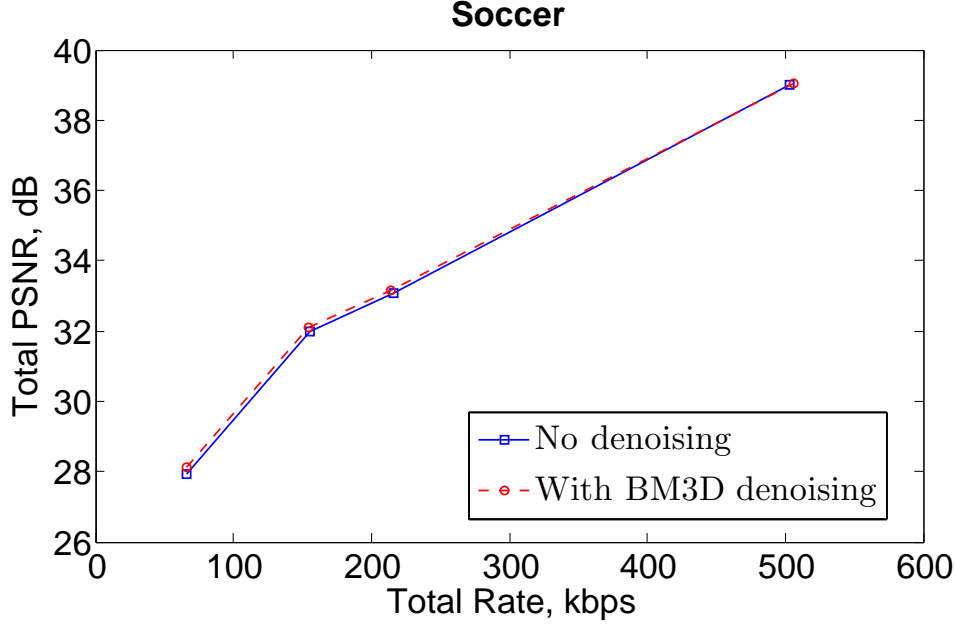


Figure 3.28: Rate-distortion performance with BM3D denoising method

olding approach, that confirms once more the hypothesis that the PSNR of the SI is not a good measure of its quality for Wyner-Ziv video decoding. It is worth noting that the BM3D solution does not require any additional information about the original image (like the true orientation map required to be known or estimated in wavelet thresholding with OWT). However, the noise variance $\hat{\sigma}_N^2$ still needs to be known or estimated.

Figure 3.28 shows the rate-distortion performance for the whole sequence (Soccer). Here the denoising is done outside the decoding cycle, with the noise variance estimated using the original sequence, which is not realistic. A small improvement can be noticed under these conditions, especially in the low rate area of the graph, where the side information has lower quality due to higher quantisation noise in key frames. Experiments in real decoding conditions, with the correlation noise estimated by the decoder, are however to be done in order to verify that this small improvement is still present.

3.6 Conclusion

In this chapter we have tried to give a global vision of the side information extraction problem in Wyner-Ziv video coding. Different ways of improving the quality of the side information have been considered.

Firstly, we have experimented with a more powerful and more precise mesh-based

motion model (instead of the block-based motion model). In conventional video coding, application of such motion models is limited due to the high bit cost of the generated motion field. In distributed video coding the complexity and the bit cost of the motion field don't matter, as the motion field need not to be transmitted. Therefore, complex motion models can be used without any limitations if they improve the rate-distortion performance of the codec.

However, by our experiments the mesh-based motion estimation has not been proved to be all the time superior to the block-based motion estimation in the distributed video coding context. On the other hand, it has been shown that it might be preferable to the latter one locally. This observation gave start to a research in the field of multiple side information combining and usage. We have presented our "hybrid" mesh/block-based approach, which slightly improves the rate-distortion performance of the DISCOVER codec. We have also considered the approach called "Multihypotheses decoding" in our study, and showed that a comparable gain can be obtained with this approach, using the same two side information hypotheses (obtained using block-based and mesh-based motion estimation respectively).

The spatial side information has also been considered in this chapter. Our experiments show that the original WZ frame of lower resolution and/or coarsely quantised (encoded at low rate with an H.264-like coder in "intra" mode), is not better than the temporal side information containing interpolation errors, but having the full spatial resolution. To explain this behaviour a hypothesis has been proposed that the channel decoder is more sensible to the spatial resolution (in this case the whole frame is affected) than to interpolation errors (which present locally in the interpolated frame). A measure SIQ of the quality of the SI has been proposed, which is more correlated with the rate-distortion performance of the codec than usually employed PSNR measure.

In our last contribution we regard the SI improvement problem as a denoising problem. We have tested two denoising methods, commonly used wavelet thresholding, and the BM3D method. Being very effective for the synthetic noise, the former method does not brings any improvement when used in the distributed video coding context. On the other hand, according to our preliminary tests the latter method allows for a slight improvement in the rate-distortion performance. However, these tests were run using the real noise variance estimated by comparing the side information with the original sequence, which is not realistic. Therefore, the first future research direction would be to verify if the denoising with the noise variance estimated by the decoder, still improves the performance.

A research direction which has been left outside of this study, is helping the side information extraction with the hash information. In fact, the authors think that transmitting a limited amount of supplementary information about the encoded WZ frame is inevitable for a good side information extraction method. In particular, this information is necessary to synchronize the position of the frame to be interpolated with the position of the WZ frame, because the motion is not always uniform in time. In

this context, the “spatial” side information from Section 3.4 can be regarded as such a hash information, and thereby be used to refine the temporal side information instead of being used directly for the decoding.

Also, looking at the gap between the performance of the realistic hybrid side information extraction method, and the performance using oracle-based SI combination mask on Figure 3.16, one can suppose that the potential of the method is not fully exploited yet. If a better correlation model estimation algorithm is found, the performance will be possibly improved significantly. The problem of correlation modeling in Wyner-Ziv coding is addressed in the next chapter.

Chapter 4

Correlation modeling in Wyner-Ziv video coding

Correlation modeling is another (after the side information extraction problem) very important problem in Wyner-Ziv coding of video. The difficulty is related to the nature of video data, which is very diverse and unstationary. Consequently, the correlation cannot be modeled precisely with a single and stationary model, and often requires a set of changing and adapting models.

Brites *et al.* [BAP06c, BAP06b, BP08] have developed a solution for on-line parameters estimation that is used in most of the latest publications on Wyner-Ziv video coding (see Section 2.3.4.6 for more details). To the best of our knowledge it is the only realistic solution existing which does require neither the knowledge of the original frame by the decoder, nor the knowledge (or estimation) of the side information at the encoder. However, as we have mentioned in Section 2.3.4.6, it assumes the distribution of the correlation noise to be similar to the distribution of the motion compensation error, which is not always verified in practice.

In this chapter, we consider different problems related to the correlation noise modeling in Wyner-Ziv video coding. In the first part, we deal with the classical Stanford architecture [ARG04, GARRM05], where only the decoder knows and exploits the correlation between the source X and the side information Y . We start by clarifying the correlation model used in the DISCOVER codec (following the Stanford's architecture), and analysing the performance of the algorithm proposed by Brites *et al.* Then, we present our algorithm of updating the correlation model during the decoding using a sort of Expectation-Minimisation (EM) algorithm. The problem of an optimal in terms of mean-squared error (MSE) reconstruction is then addressed, and closed-form expressions for the Laplacian correlation model are derived.

The second part extends the study of the correlation to the encoder side. Firstly, we propose an algorithm for the minimal rate estimation to reduce the number of decoder requests of parity bits. This algorithm works on the encoder side, but the correlation

parameter can be estimated and transmitted from the decoder in on-line applications (e.g. using the algorithm in [BP08]). After that we discuss the algorithm proposed in [Laj06] and adopted in DISCOVER for the turbo decoder bit error rate estimation, and propose an improvement. In fact, the two above problems do not concern only the encoder, but require a cooperation between the encoder and the decoder. That is why the proposed solution is sometimes called “Hybrid rate-control” (to distinguish both from encoder and decoder rate control).

In the last third part of the chapter, we propose several algorithms to improve the rate-distortion performance by exploiting the knowledge of the correlation at the encoder. First, we study the rate-distortion optimisation problem in Wyner-Ziv video coding. An algorithm for an optimal in terms of rate-distortion performance (for the assumed correlation model) bit plane allocation is proposed, which takes the decision based on the estimated rate and distortion. This algorithm, however, requires a precise estimation of the correlation model parameters. According to our experiments, the parameters estimated at the decoder side using the algorithm of Brites *et al.* are not suitable. Therefore, we propose a robust estimation algorithm at the encoder side, performing the motion estimation for a limited part of the frame. Having a downside of adding complexity to the encoder, the approach allows to better estimate the correlation model parameter.

Second, we propose a mixed “intra”/WZ coding mode for WZ frames. The idea is inspired by the PRISM architecture where different encoding modes can be selected for different blocks of the image. Experimental results show significant gains in rate-distortion performance for sequences with high motion, compared to the DISCOVER codec with fixed GOP size 2.

Finally, a conclusion on the problem of correlation modeling for Wyner-Ziv video coding, is made, and several future work directions in this field are outlined.

4.1 DISCOVER correlation model

Let us introduce some notations first. In this chapter, we hold to the DISCOVER architecture described in Section 2.3.4. To be consistent with Section 2.3.4, let X_B and X_F denote the backward and the forward reference frames respectively (decoded key frames or previously decoded WZ frames), and X denotes the original encoded frame. If $\mathbf{u}(\mathbf{s}) = (\mathbf{u}_B(\mathbf{s}), \mathbf{u}_F(\mathbf{s}))$ denotes the bidirectional motion vector for every point \mathbf{s} in the frame X , then the side information Y and the motion compensated residual frame R are defined as follows (see Section 2.3.4.5):

$$Y(\mathbf{s}) = \frac{t_F X_B(\mathbf{s} + \mathbf{u}_B) + t_B X_F(\mathbf{s} + \mathbf{u}_F)}{t_F + t_B}, \quad (4.1)$$

$$R(\mathbf{s}) = \frac{X_B(\mathbf{s} + \mathbf{u}_B) - X_F(\mathbf{s} + \mathbf{u}_F)}{2}, \quad (4.2)$$

where t_F and t_B are the distances in the temporal domain from X to X_B and X_F respectively (see Figure 2.20).

Let $y_{k,i}$, $r_{k,i}$ and $x_{k,i}$ denote DCT coefficients in the frequency band k , $k = 1, \dots, 16$, at the position i of transformed images Y , R and X respectively.

Two levels of correlation modeling are proposed in [BAP06c, BAP06b, BP08]. The coarse level is the band level, with a single Laplacian model $\mathcal{L}(0, \alpha_k)$ for the correlation noise $z_{k,i} = x_{k,i} - y_{k,i}$ at all positions i of the band:

$$p_{x|y}(x) = \frac{\alpha_k}{2} e^{-\alpha_k |x - y_{k,i}|}. \quad (4.3)$$

The second, finest level of correlation modeling is the coefficient level. With this approach the correlation for each coefficient position i is modeled independently of the others, with its own parameter $\alpha_{k,i}$:

$$p_{x|y}(x) = \frac{\alpha_{k,i}}{2} e^{-\alpha_{k,i} |x - y_{k,i}|}. \quad (4.4)$$

Let us denote with $\hat{\alpha}_k$ and $\hat{\alpha}_{k,i}$ the values of these parameters estimated with the algorithm in [BP08]:

$$\hat{\alpha}_k = \sqrt{\frac{2}{\hat{\sigma}_k^2}}, \quad (4.5)$$

$$\hat{\alpha}_{k,i} = \begin{cases} \hat{\alpha}_k & \text{if } [|r_{k,i}| - \hat{\mu}_{|k|}]^2 \leq \hat{\sigma}_{|k|}^2, \\ \sqrt{\frac{2}{[|r_{k,i}| - \hat{\mu}_{|k|}]^2}} & \text{if } [|r_{k,i}| - \hat{\mu}_{|k|}]^2 > \hat{\sigma}_{|k|}^2, \end{cases} \quad (4.6)$$

where $\hat{\mu}_{|k|}$ and $\hat{\sigma}_{|k|}^2$ are the mean and the variance of *absolute values* of residual DCT coefficients $|r_{k,i}|$, and $\hat{\sigma}_k^2$ is the variance of residual DCT coefficients $r_{k,i}$ (see Section 2.3.4.6).

For experimental purposes, the “true” correlation model can also be derived using the original frame X and the side information Y . Let us denote “true” values of model parameters corresponding to that model, with α_k^T and $\hat{\alpha}_{k,i}^T$:

$$\alpha_k^T = \sqrt{\frac{2}{\sigma_k^2}} = \sqrt{\frac{2}{\frac{1}{N_k} \sum_i [x_{k,i} - y_{k,i} - \frac{1}{N_k} \sum_j (x_{k,j} - y_{k,j})]^2}}, \quad (4.7)$$

$$\hat{\alpha}_{k,i}^T = \frac{1}{|x_{k,i} - y_{k,i}|} \quad (4.8)$$

(the “true” value for $\hat{\alpha}_{k,i}^T$ has been chosen so that the expectation of the absolute value of the correlation noise $|Z| = |X - Y|$ is equal to $|x_{k,i} - y_{k,i}|$).

The relationship between $\alpha_{k,i}^T$, α_k^T , $\hat{\alpha}_{k,i}$ and $\hat{\alpha}_k$ is shown schematically on Figure 4.1. On this figure, the “estimation error” arrow represents a possible mismatch between

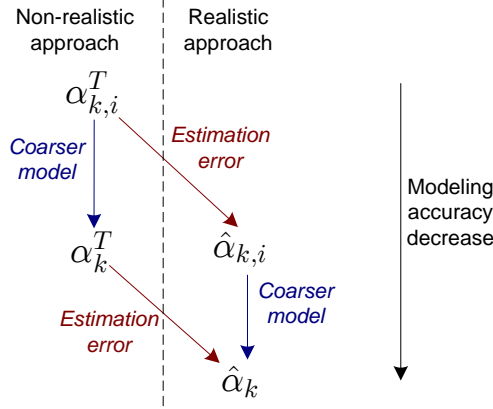


Figure 4.1: Relationship between different correlation models

$|r_{k,i}|$ and $|x_{k,i} - y_{k,i}|$ distributions (locally as well as globally), while the “coarser model” arrow represents the transition from the local model (on the coefficients’ level) to the coarser global model with a single parameter. Obviously, the accuracy of the model is decreased by each of these two factors.

Figure 4.2 shows an example of correlation noise histograms with imposed global models (estimated and the “true” one) for bands 1 and 3 of the frame # 2 of the Foreman sequence. This example shows a typical relationship ($\hat{\alpha}_k \geq \alpha_k^T$) between α_k^T and $\hat{\alpha}_k$. In other words, the global correlation model parameter $\hat{\alpha}_k$ is often overestimated by the method of Brites *et al.* [BP08]. It can be explained by an observation that the “true” correlation noise $|x_{k,i} - y_{k,i}|$ is not very likely to be smaller than the interpolation error $|r_{k,i}|$, while it can easily be much higher (e.g. if the side information extraction totally fails for some blocks, or when the motion is not uniform).

However, this slight over-estimation plays a positive role when the coefficient-based correlation model is used, because the global over-estimated parameter $\hat{\alpha}_k$ is used only for the coefficients close to zero. Indeed, the “true” global model with α_k^T does not suite very well those coefficients (see histograms on Figures 4.2(a) and 4.2(b)) because of the presence of outliers of the model (very high values $|x_{k,i} - y_{k,i}|$). At the same time, the over-estimated model with $\hat{\alpha}_k$ better suits the histogram of the “true” correlation noise $|x_{k,i} - y_{k,i}|$ around zero.

Figure 4.3 shows the rate-distortion performance of the DISCOVER codec with different correlation models employed. It is clear that the knowledge of the “true” correlation per DCT coefficient ($\alpha_{k,i}^T$) gives a great advantage to the decoder with respect to other models. Possibly, the search for new correlation models and/or for more robust methods for the estimation of their parameters is one of the most promising research directions in Wyner-Ziv video coding.

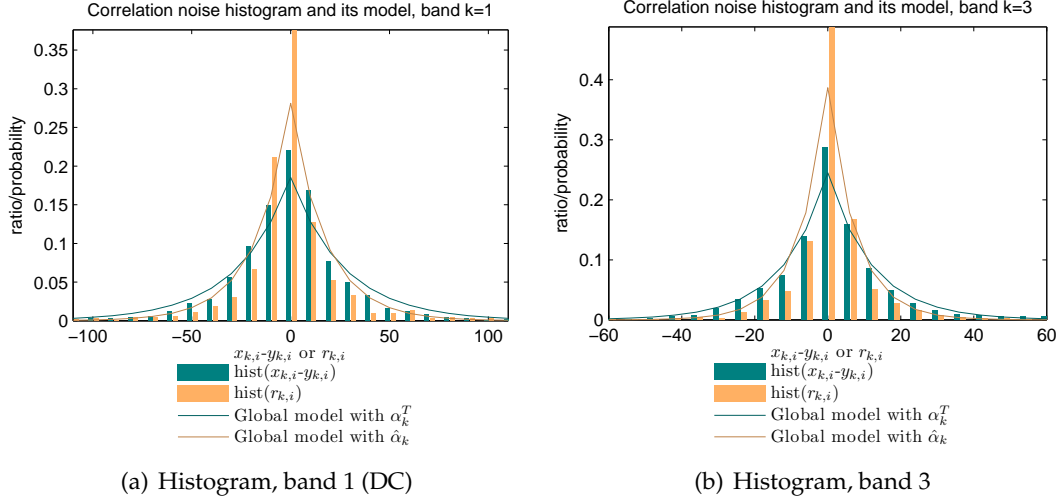


Figure 4.2: Example of correlation models for the frame #2 of Foreman, QCIF

4.2 Updating the correlation model during decoding

As we showed in the previous section, there is a gap between the codec performance with “true” correlation model parameters and with estimated ones (especially for the local model). In order to try to reduce this gap, a method for the on-line (during decoding) parameter learning was developed for the version of the DISCOVER codec using turbo codes for channel coding.

We start the study on model learning with the simpler case of the global model. Informally speaking, the goal is to learn α_k^T while the decoding starts with $\hat{\alpha}_k$. For this purpose, a variation of the Expectation-Maximisation (EM) algorithm is employed. The description of the EM algorithm can be found in [Moo96] for example.

Let Θ denote the model parameter to be learnt (α_k^T), and $P_{\Theta}(\mathbf{x}_k, \mathbf{y}_k)$ is the likelihood measure of the Laplacian correlation model with parameter Θ :

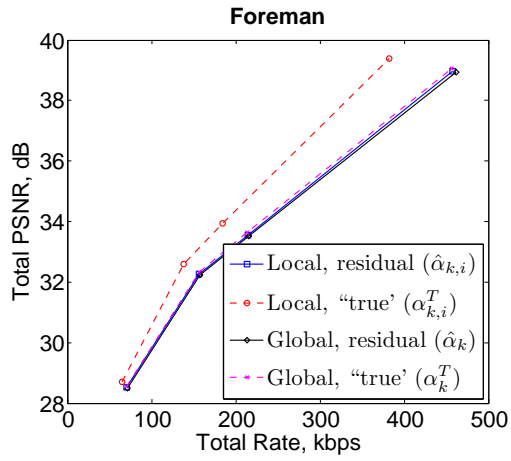
$$P_{\Theta}(\mathbf{x}_k, \mathbf{y}_k) = \Pr_{\Theta}(\mathbf{x}_k | \mathbf{y}_k) = \prod_i \frac{\Theta}{2} e^{-\Theta |x_{k,i} - y_{k,i}|}. \quad (4.9)$$

We will omit the band index k in the remaining part of this section.

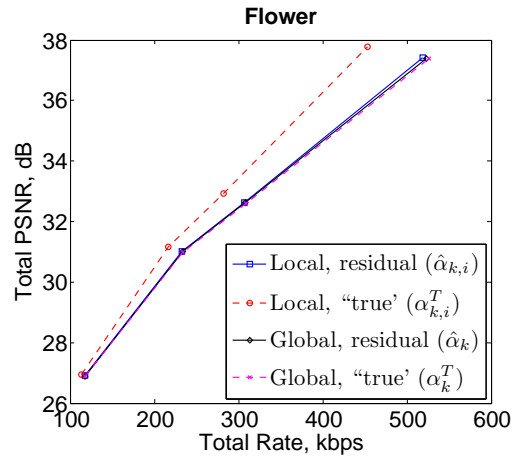
The EM algorithm consists in iteratively improving the initial estimation Θ_0 of the parameter Θ (in our case $\Theta_0 = \hat{\alpha}_k$) by alternating expectation and maximisation steps.

The E-step (Expectation) of the EM algorithm consists in calculating the expectation of the log-likelihood ratio (LLR):

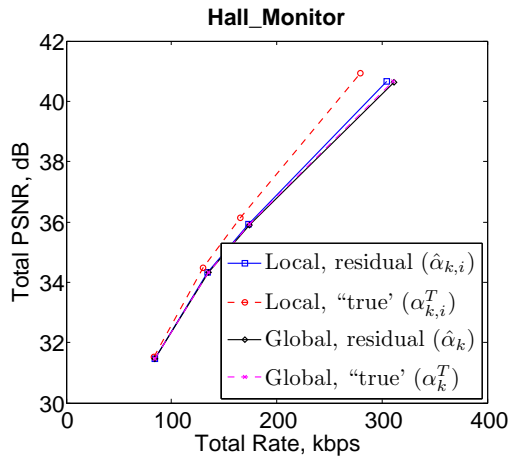
$$\log \frac{P_{\Theta_n}(\mathbf{x}, \mathbf{y})}{P_{\Theta}(\mathbf{x}, \mathbf{y})} \quad (4.10)$$



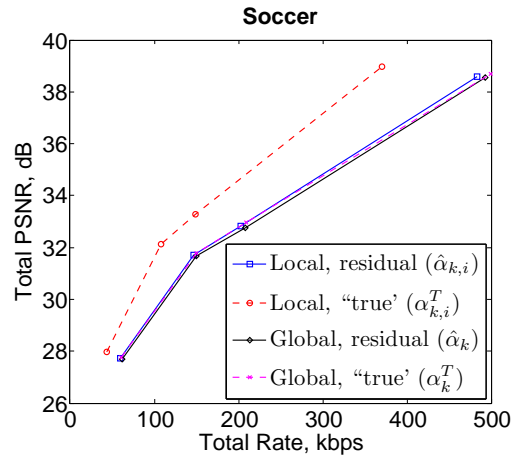
(a) Foreman sequence, QCIF, 15Hz



(b) FlowerGarden sequence, QCIF, 15Hz



(c) Hall monitor sequence, QCIF, 15Hz



(d) Soccer sequence, QCIF, 15Hz

Figure 4.3: Rate-distortion performance with different correlation models

with respect to \mathbf{x} (not available at the decoder), which is modeled using the current value Θ_n :

$$\mathbb{E}_{\mathbf{x}, \Theta_n} \left[\log \frac{P_{\Theta_n}(\mathbf{x}, \mathbf{y})}{P_{\Theta}(\mathbf{x}, \mathbf{y})} \middle| \mathbf{y} \right] = \int_{\mathbf{x}} \log \frac{P_{\Theta_n}(\mathbf{x}, \mathbf{y})}{P_{\Theta}(\mathbf{x}, \mathbf{y})} \mathbf{p}_{\Theta_n}(\mathbf{x}|\mathbf{y}) d\mathbf{x}, \quad (4.11)$$

where $\mathbf{p}_{\Theta_n}(\mathbf{x}|\mathbf{y})$ is a conditional probability density function (p.d.f.) of \mathbf{x} knowing \mathbf{y} :

$$\mathbf{p}_{\Theta_n}(\mathbf{x}|\mathbf{y}) = \prod_i \frac{\Theta_n}{2} e^{-\Theta_n |x_i - y_i|}. \quad (4.12)$$

The next estimation Θ_{n+1} is given by maximising the expectation (4.11) taken with the negative sign (the M-step):

$$\Theta_{n+1} = \arg \max_{\Theta} \left(-\mathbb{E}_{\mathbf{x}, \Theta_n} \left[\log \frac{P_{\Theta_n}(\mathbf{x}, \mathbf{y})}{P_{\Theta}(\mathbf{x}, \mathbf{y})} \middle| \mathbf{y} \right] \right) \quad (4.13)$$

For this maximisation, we can ignore the numerator of the LLR as it does not depend on Θ . Thereby we get:

$$\Theta_{n+1} = \arg \max_{\Theta} \int_{\mathbf{x}} \log P_{\Theta}(\mathbf{x}, \mathbf{y}) \mathbf{p}_{\Theta_n}(\mathbf{x}|\mathbf{y}) d\mathbf{x} \quad (4.14)$$

Using (4.9), we obtain:

$$\log P_{\Theta}(\mathbf{x}, \mathbf{y}) = \sum_i \log \left[\frac{\Theta}{2} e^{-\Theta |x_i - y_i|} \right] = \sum_i \log \frac{\Theta}{2} - \Theta \sum_i |x_i - y_i|, \quad (4.15)$$

and consequently

$$\int_{\mathbf{x}} \log P_{\Theta}(\mathbf{x}, \mathbf{y}) \mathbf{p}_{\Theta_n}(\mathbf{x}|\mathbf{y}) d\mathbf{x} = N \log \frac{\Theta}{2} \underbrace{\int_{\mathbf{x}} \mathbf{p}_{\Theta_n}(\mathbf{x}|\mathbf{y}) d\mathbf{x}}_{=1} - \Theta \underbrace{\int_{\mathbf{x}} \sum_i |x_i - y_i| \mathbf{p}_{\Theta_n}(\mathbf{x}|\mathbf{y}) d\mathbf{x}}_{=\mathbb{E}_{\mathbf{x}, \Theta_n} [\sum_i |x_i - y_i| | \mathbf{y}]}, \quad (4.16)$$

where N is the number of coefficients in a band.

Finally:

$$\Theta_{n+1} = \arg \max_{\Theta} \left(N \log \frac{\Theta}{2} - \Theta \sum_i \mathbb{E}_{\mathbf{x}, \Theta_n} [|x_i - y_i| | \mathbf{y}] \right) = \arg \max_{\Theta} Q(\Theta, \Theta_n). \quad (4.17)$$

Taking the derivative of $Q(\Theta, \Theta_n)$ with respect to Θ :

$$\frac{\partial Q(\Theta, \Theta_n)}{\partial \Theta} = \frac{N}{\Theta} - \sum_i \mathbb{E}_{\mathbf{x}, \Theta_n} [|x_i - y_i| | \mathbf{y}], \quad (4.18)$$

the optimal value Θ_{n+1} maximising the expectation (4.11) can be found as the solution of $\frac{\partial Q(\Theta, \Theta_n)}{\partial \Theta} = 0$:

$$\Theta_{n+1} = \frac{N}{\sum_i \mathbb{E}_{\mathbf{x}, \Theta_n} [|x_i - y_i| | \mathbf{y}]}. \quad (4.19)$$

Now, the value of $\mathbb{E}_{\mathbf{x}, \Theta_n} [|x_i - y_i| | \mathbf{y}] = \mathbb{E}_{x_i, \Theta_n} [|x_i - y_i| | y_i]$ is to be calculated. We use turbo decoder extrinsic probabilities in the calculation, which is explained below.

Let b be the number of the bit plane which is being currently decoded, $1 \leq b \leq M_k$, and $I^{b-1}(x_i^1, \dots, x_i^{b-1})$ denote the quantisation interval defined by $(b-1)$ most significant bits of x_i (already decoded by this moment), I^0 being the range of all possible values of X . Please note that in general $I^{b-1}(x_i^1, \dots, x_i^{b-1})$, $b \geq 2$, can be a set of disjoint intervals (it depends on the mapping of quantisation indices). The decoder extrinsic information is the extrinsic probability of x_i of being inside one of the intervals $I_0 = I^b(x_i^1, \dots, x_i^{b-1}, 0)$ (we denote this probability as p_i^{ext}) or $I_1 = I^b(x_i^1, \dots, x_i^{b-1}, 1)$ (denoted respectively as $1 - p_i^{ext}$).

The expectation $\mathbb{E}_{x_i, \Theta_n} [|x_i - y_i| | y_i]$ is then computed as follows:

$$\mathbb{E}_{x_i, \Theta_n} [|x_i - y_i| | y_i] = \frac{p_i^{ext} \int_{I_0} |x - y_i| \frac{\Theta_n}{2} e^{-\Theta_n |x - y_i|} dx + (1 - p_i^{ext}) \int_{I_1} |x - y_i| \frac{\Theta_n}{2} e^{-\Theta_n |x - y_i|} dx}{p_i^{ext} \int_{I_0} \frac{\Theta_n}{2} e^{-\Theta_n |x - y_i|} dx + (1 - p_i^{ext}) \int_{I_1} \frac{\Theta_n}{2} e^{-\Theta_n |x - y_i|} dx}. \quad (4.20)$$

Given that for any continuous $[a, b]$:

$$\int_a^b |x - y| \frac{\alpha}{2} e^{-\alpha |x - y|} dx = \begin{cases} \left| \frac{|y - a| + \frac{1}{\alpha}}{2} e^{-\alpha |y - a|} - \frac{|y - b| + \frac{1}{\alpha}}{2} e^{-\alpha |y - b|} \right|, & \text{if } y < a \vee y > b, \\ \frac{1}{\alpha} - \frac{y - a + \frac{1}{\alpha}}{2} e^{-\alpha(y - a)} - \frac{b - y + \frac{1}{\alpha}}{2} e^{-\alpha(b - y)}, & \text{if } a \leq y \leq b, \end{cases} \quad (4.21)$$

and

$$\int_a^b \frac{\alpha}{2} e^{-\alpha |x - y|} dx = \begin{cases} \left| e^{-\alpha |y - a|} - e^{-\alpha |y - b|} \right|, & \text{if } y < a \vee y > b, \\ 1 - \frac{e^{-\alpha(y - a)} + e^{-\alpha(b - y)}}{2}, & \text{if } a \leq y \leq b, \end{cases} \quad (4.22)$$

integrals in (4.20) over I_0 and I_1 can be easily calculated (by subdividing into continuous subintervals if necessary).

Figure 4.4 shows the scheme of the turbo decoder with on-line correlation model parameter learning using the above algorithm (denoted as “EM”). Note that the learning module can be put inside the internal turbo decoding cycle, but we don’t do it as it would increase the decoder complexity significantly (because channel probabilities have to be recalculated after each parameter update). Instead, we do several sub-iterations of (4.19) with the same extrinsic information, only updating Θ_{n+1} on each sub-iteration.

Figure 4.5 shows an example of parameter learning with the above algorithm for bands 1 and 3 of the frame #2 of the Foreman sequence. On the horizontal axis we plotted the global number n of the algorithm iteration per band. We have fixed the number of sub-iterations (see above) to 5, so the total number of iterations equals to the total number of requests per band (including all bit planes) multiplied by 5. The decoding is started by the most significant bit plane (MSB).

In this test the decoder uses the global correlation model estimated using residual DCT coefficients $r_{k,i}$, therefore it starts with \hat{a}_k . As the decoding of the band advances

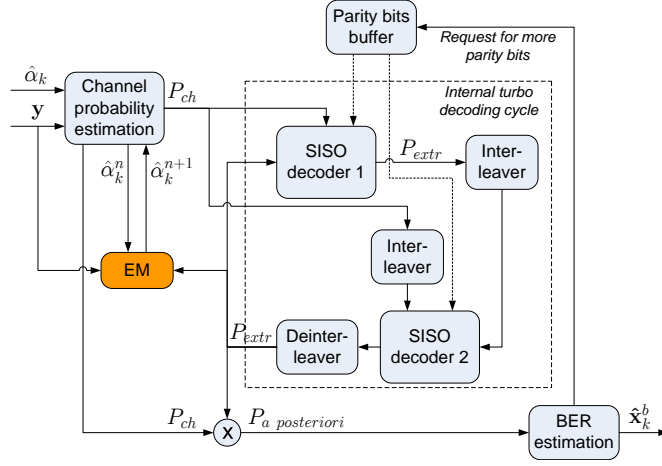


Figure 4.4: Turbo decoding with on-line parameter learning

(e.g. more parity bits are received), the learnt value Θ_n becomes closer to the “true” value α_k^T (see Figure 4.5), thus improving the estimation of channel probabilities for remaining bit planes as well as the final reconstruction quality of the band.

Now we extend the above approach to include the possibility of refining the local correlation model (i.e. per DCT coefficient), defined by (4.6). Let $\Theta_{n,i}$ denote the correlation model parameter for the coefficient at the position i , and $\Theta_{0,i} = \hat{\alpha}_{k,i}$. The formula (4.19) is then transformed into:

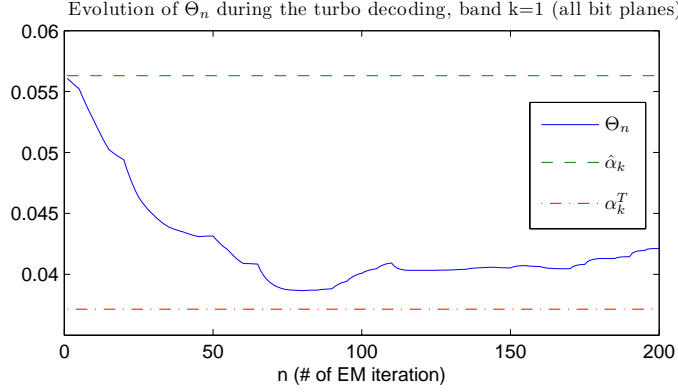
$$\Theta_{n+1} = \frac{N}{\sum_i \mathbb{E}_{x_i, \Theta_{n,i}} [|x_i - y_i| |y_i|]}. \quad (4.23)$$

Similarly to (4.6) we define:

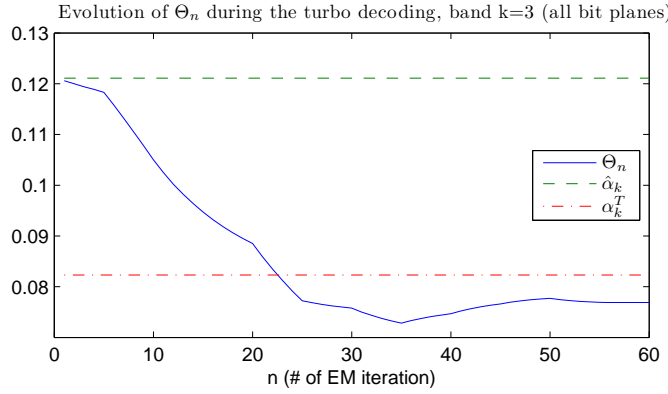
$$\Theta_{n+1,i} = \begin{cases} \Theta_{n+1} & \text{if } [|r_{k,i}| - \hat{\mu}_{|k|}]^2 \leq \hat{\sigma}_{|k|}^2, \\ \sqrt{\frac{2}{[|r_{k,i}| - \hat{\mu}_{|k|}]^2}} & \text{if } [|r_{k,i}| - \hat{\mu}_{|k|}]^2 > \hat{\sigma}_{|k|}^2, \end{cases} \quad (4.24)$$

thus only the coefficients which are modeled using the global model are affected by the model learning algorithm.

The rate-distortion performance of the DISCOVER codec with the integrated correlation model parameter learning algorithm (marked as “+ EM”) is shown on Figure 4.6. For all sequences, the learning technique slightly improves the performance with the *global* correlation model. At the same time, the performance with the *local* model is not always improved by using the proposed learning technique. This may happen due to the fact that the local model itself does not correspond well to the real data. For the “Hall monitor” sequence, which is easier for modeling, the learning approach improves the performance in both cases.



(a) Band 1 (DC)



(b) Band 3

Figure 4.5: Correlation model parameter learning during turbo decoding

4.3 Minimal MSE reconstruction

The work described in this section has been done jointly with Jayanth Nayak. The results have been published in the conference paper [KNG07], which has been awarded a “Best Student Paper Award”.

The minimal mean-squared error (MSE) reconstruction value of a DCT coefficient $\hat{x}_{k,i}$ is given by the formula:

$$\hat{x}_{k,i}^{mmse} = \mathbb{E}_{p_{x|y}} [x | x \in [l, u), y_{k,i}], \quad (4.25)$$

where $[l, u)$ is the quantisation interval for $x_{k,i}$ (assumed known as the Slepian-Wolf coding part is supposed to be lossless), $y_{k,i}$ is the side information DCT coefficient, and the expectation is taken over the conditional p.d.f. $p_{x|y}(x_{k,i}|y_{k,i})$ of X knowing Y :

$$\hat{x}_{k,i}^{mmse} = \mathbb{E}_{p_{x|y}} [x | x \in [l, u), y_{k,i}] = \frac{\int_l^u x p_{x|y_{k,i}} dx}{\int_l^u p_{x|y_{k,i}} dx}. \quad (4.26)$$

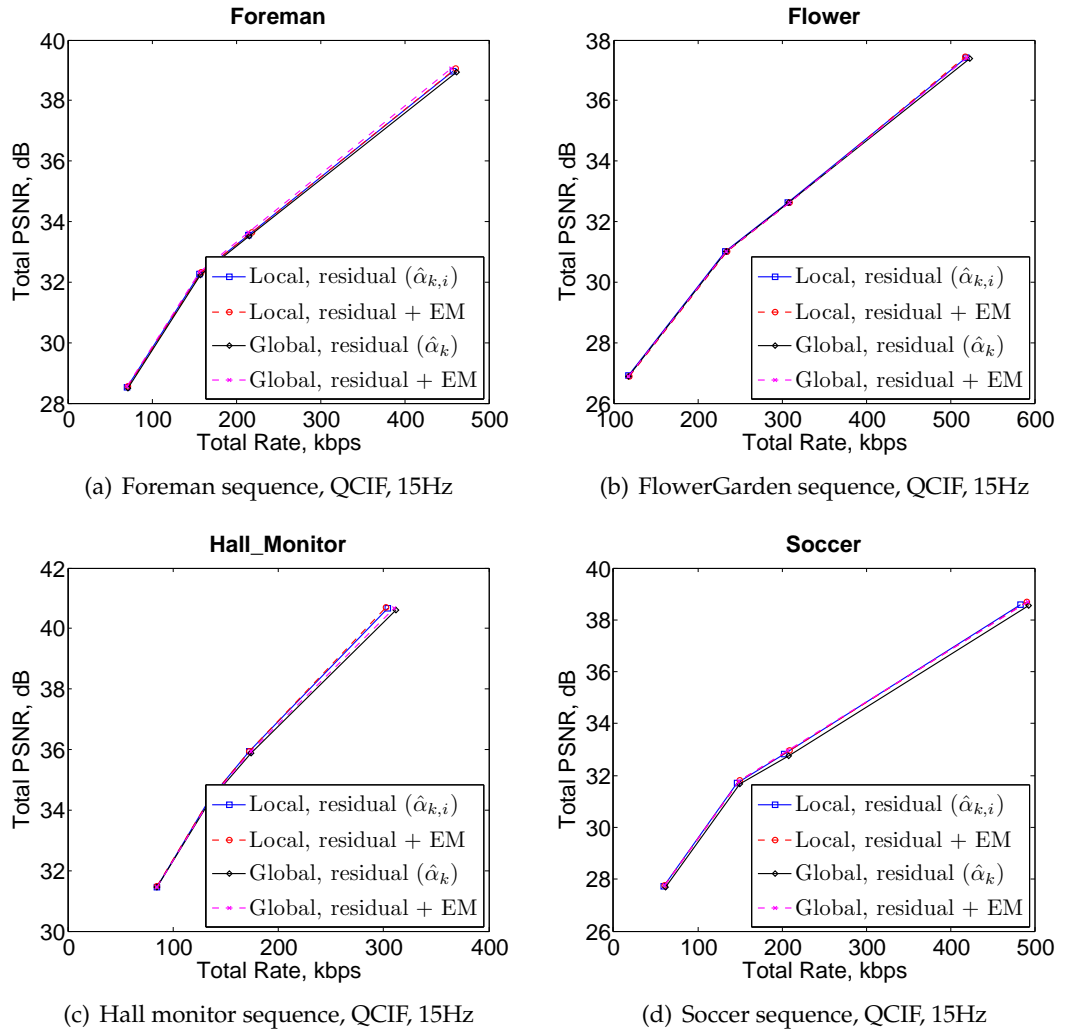


Figure 4.6: Rate-distortion performance with correlation model learning

However, until recently many Wyner-Ziv video codec implementations in literature have been using sub-optimal solutions, partly because of the fact that no closed-form expressions were derived even for the commonly adopted Laplacian correlation model. For example, the IST Wyner-Ziv video coding software the DISCOVER codec is based on, used the following reconstruction approach, proposed in [AZG02]:

$$\hat{x}_{k,i} = \begin{cases} l, & y_{k,i} < l \\ y_{k,i}, & y_{k,i} \in [l, u) \\ u, & y_{k,i} \geq u \end{cases} . \quad (4.27)$$

Below we will refer to this reconstruction approach as to the “straightforward reconstruction” method. In 2007, Vatis *et al.* [VKO07] have proposed another approach optimised on a large training data set. Although it is an empirical solution, its performance is very close to the performance of the MSE-optimal reconstruction method.

In this section we show how to derive closed-form expressions for (4.26), thereby providing an efficient method for the MSE-optimal reconstruction for the Laplacian correlation model. We also extend the derivation to the multihypotheses decoding, which has been described in Section 3.3.

4.3.1 Closed-form expressions for the single side information case

For the Laplacian correlation model we have:

$$p_{x|y}(x|y_{k,i}) = \frac{\alpha_{k,i}}{2} e^{-\alpha_{k,i}|x-y_{k,i}|}, \quad (4.28)$$

where $\alpha_{k,i}$ is a local model parameter ($\alpha_{k,i} = \alpha_k \forall i$, if the global correlation model is used). We omit indices k and i below for brevity.

Therefore:

$$\hat{x}^{mmse} = \frac{\int_l^u x p_{x|y} dx}{\int_l^u p_{x|y} dx} = \frac{\int_l^u \frac{\alpha}{2} x e^{-\alpha|x-y|} dx}{\int_l^u \frac{\alpha}{2} e^{-\alpha|x-y|} dx}. \quad (4.29)$$

Let us first calculate the integral in the denominator. To get rid of the modulus operation, three different cases are distinguished:

- $y < l$:

$$\int_l^u \frac{\alpha}{2} e^{-\alpha|x-y|} dx = \int_l^u \frac{\alpha}{2} e^{-\alpha(x-y)} dx = \frac{\alpha}{2} \int_l^u \frac{1}{-\alpha} d(e^{-\alpha(x-y)}) = -\frac{1}{2} (e^{-\alpha(u-y)} - e^{-\alpha(l-y)}). \quad (4.30)$$

- $y > u$:

$$\int_l^u \frac{\alpha}{2} e^{-\alpha|x-y|} dx = \int_l^u \frac{\alpha}{2} e^{-\alpha(y-x)} dx = \frac{\alpha}{2} \int_l^u \frac{1}{\alpha} d(e^{-\alpha(y-x)}) = \frac{1}{2} (e^{-\alpha(y-u)} - e^{-\alpha(y-l)}). \quad (4.31)$$

- $y \in [l, u]$:

$$\int_l^u \frac{\alpha}{2} e^{-\alpha|x-y|} dx = \int_l^y \frac{\alpha}{2} e^{-\alpha(y-x)} dx + \int_y^u \frac{\alpha}{2} e^{-\alpha(x-y)} dx. \quad (4.32)$$

Please note that these two integrals are similar to (4.31) with $u = y$ and (4.30) with $l = y$ respectively. Thus (4.32) can be written as:

$$\int_l^u \frac{\alpha}{2} e^{-\alpha|x-y|} dx = \frac{1}{2} (e^{-\alpha(y-y)} - e^{-\alpha(y-l)}) - \frac{1}{2} (e^{-\alpha(u-y)} - e^{-\alpha(y-y)}) = 1 - \frac{e^{-\alpha(y-l)} + e^{-\alpha(u-y)}}{2}. \quad (4.33)$$

Finally:

$$\int_l^u \frac{\alpha}{2} e^{-\alpha|x-y|} dx = \begin{cases} \frac{e^{-\alpha(l-y)} - e^{-\alpha(u-y)}}{2}, & \text{if } y < l, \\ 1 - \frac{e^{-\alpha(y-l)} + e^{-\alpha(u-y)}}{2}, & \text{if } y \in [l, u], \\ \frac{e^{-\alpha(y-u)} - e^{-\alpha(y-l)}}{2}, & \text{if } y > u. \end{cases} \quad (4.34)$$

The integral in the numerator of (4.29) is computed in a similar way:

- $y < l$:

$$\int_l^u \frac{\alpha}{2} x e^{-\alpha|x-y|} dx = \int_l^u \frac{\alpha}{2} x e^{-\alpha(x-y)} dx = \frac{\alpha}{2} \int_l^u \frac{1}{-\alpha} x d(e^{-\alpha(x-y)}). \quad (4.35)$$

Integrating by parts, we obtain:

$$\begin{aligned} \frac{\alpha}{2} \int_l^u \frac{1}{-\alpha} x d(e^{-\alpha(x-y)}) &= -\frac{1}{2} [x e^{-\alpha(x-y)}]_{x=l}^{x=u} + \frac{1}{\alpha} \overbrace{\int_l^u \frac{\alpha}{2} e^{-\alpha(x-y)}}^{\text{given by (4.30)}} \\ &= -\frac{1}{2} [u e^{-\alpha(u-y)} - l e^{-\alpha(l-y)}] + \frac{e^{-\alpha(l-y)} - e^{-\alpha(u-y)}}{2\alpha} \\ &= \frac{(1 + \alpha l) e^{-\alpha(l-y)} - (1 + \alpha u) e^{-\alpha(u-y)}}{2\alpha}. \end{aligned} \quad (4.36)$$

- $y < l$:

$$\begin{aligned} \int_l^u \frac{\alpha}{2} x e^{-\alpha|x-y|} dx &= \int_l^u \frac{\alpha}{2} x e^{-\alpha(y-x)} dx = \frac{\alpha}{2} \int_l^u \frac{1}{\alpha} x d(e^{-\alpha(y-x)}) \\ &= \frac{1}{2} [x e^{-\alpha(y-x)}]_{x=l}^{x=u} - \frac{1}{\alpha} \overbrace{\int_l^u \frac{\alpha}{2} e^{-\alpha(y-x)}}^{\text{given by (4.31)}} \\ &= \frac{1}{2} [u e^{-\alpha(y-u)} - l e^{-\alpha(y-l)}] - \frac{e^{-\alpha(y-u)} - e^{-\alpha(y-l)}}{2\alpha} \\ &= \frac{(1 - \alpha l) e^{-\alpha(y-l)} - (1 - \alpha u) e^{-\alpha(y-u)}}{2\alpha}. \end{aligned} \quad (4.37)$$

- $y \in [l, u]$:

$$\begin{aligned}
\int_l^u \frac{\alpha}{2} x e^{-\alpha|x-y|} dx &= \overbrace{\int_l^y \frac{\alpha}{2} x e^{-\alpha(y-x)} dx}^{\text{given by (4.37) with } u=y} + \overbrace{\int_y^u \frac{\alpha}{2} x e^{-\alpha(x-y)} dx}^{\text{given by (4.36) with } l=y} \\
&= \frac{(1-\alpha l)e^{-\alpha(y-l)} - (1-\alpha y) + (1+\alpha y) - (1+\alpha u)e^{-\alpha(u-y)}}{2\alpha} \\
&= y - \frac{(1-\alpha l)e^{-\alpha(y-l)} - (1+\alpha u)e^{-\alpha(u-y)}}{2\alpha}. \tag{4.38}
\end{aligned}$$

The numerator can then be computed as follows:

$$\int_l^u \frac{\alpha}{2} x e^{-\alpha|x-y|} dx = \begin{cases} \frac{(1+\alpha l)e^{-\alpha(l-y)} - (1+\alpha u)e^{-\alpha(u-y)}}{2\alpha}, & \text{if } y < l, \\ y - \frac{(1-\alpha l)e^{-\alpha(y-l)} - (1+\alpha u)e^{-\alpha(u-y)}}{2\alpha}, & \text{if } y \in [l, u], \\ \frac{(1-\alpha l)e^{-\alpha(y-l)} - (1-\alpha u)e^{-\alpha(y-u)}}{2\alpha}, & \text{if } y > u. \end{cases} \tag{4.39}$$

Putting together (4.39) and (4.34), the minimal MSE reconstruction $\hat{x}_{k,i}^{mmse}$ (4.26) is given by:

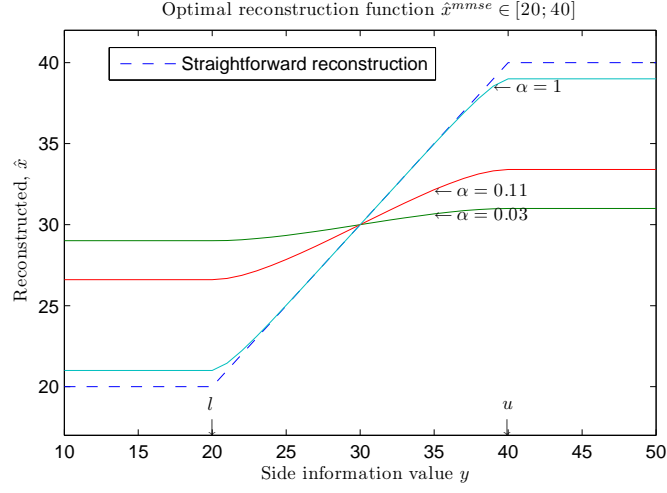
$$\hat{x}^{mmse} = \begin{cases} l + \frac{1}{\alpha} + \frac{u-l}{1-e^{\alpha(u-l)}}, & \text{if } y < l, \\ y + \frac{\left(y-l + \frac{1}{\alpha}\right)e^{-\alpha(y-l)} - \left(u-y + \frac{1}{\alpha}\right)e^{-\alpha(u-y)}}{2 - (e^{-\alpha(y-l)} + e^{-\alpha(u-y)})}, & \text{if } y \in [l, u], \\ u - \frac{1}{\alpha} - \frac{u-l}{1-e^{\alpha(u-l)}}, & \text{if } y > u. \end{cases} \tag{4.40}$$

Figure 4.7 shows several MSE-optimal reconstruction functions given by (4.40) for x in $[20, 40]$ for different values of α . We note that when the correlation increases (α increases), the optimal approach tends to the straightforward reconstruction approach (4.27), the two methods being equivalent at $\alpha = \infty$. The rate-distortion performance results of the proposed optimal reconstruction method will be presented below, after the extension to the multihypotheses decoding.

4.3.2 Closed-form expressions for multihypotheses decoding

Multihypotheses decoding approach is described in Section 3.3. It consists in considering two side information hypotheses $y_{k,i}^1$ and $y_{k,i}^2$, each having its own correlation model with parameters $\alpha_{k,i}^1$ and $\alpha_{k,i}^2$. The MSE-optimal reconstruction in this case is given by:

$$\hat{x}_{k,i}^{mmse, \text{MH}} = \mathbb{E}_{p_{x|y^1, y^2}} \left[x \mid x \in [l, u], y_{k,i}^1, y_{k,i}^2 \right] = \frac{\int_l^u x p_{x|y_{k,i}^1, y_{k,i}^2}(x|y^1, y^2) dx}{\int_l^u p_{x|y_{k,i}^1, y_{k,i}^2}(x|y^1, y^2) dx}, \tag{4.41}$$

Figure 4.7: Optimal reconstruction functions for different α

where $p_{x|y_{k,i}^1, y_{k,i}^2}(x|y^1, y^2)$ is the conditional p.d.f. of X knowing Y^1 and Y^2 . In fact, the conditional p.d.f. $p_{x|y_{k,i}^1, y_{k,i}^2}(x|y^1, y^2)$ is not available as the form of the correlation between Y^1 and Y^2 is not known. Instead, we use a simplified model for this p.d.f. as described in Section 3.3:

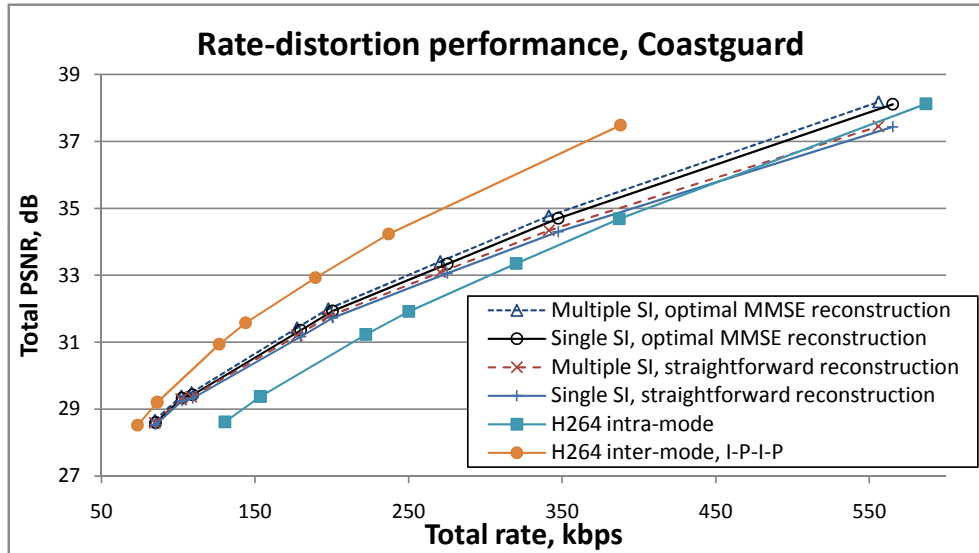
$$p_{x|y_{k,i}^1, y_{k,i}^2}(x|y^1, y^2) = \frac{p_{x|y_{k,i}^1}(x|y) + p_{x|y_{k,i}^2}(x|y)}{2}, \quad (4.42)$$

where $p_{x|y_{k,i}^1}(x|y)$ and $p_{x|y_{k,i}^2}(x|y)$ are known Laplacian conditional p.d.f. corresponding to each side information hypothesis.

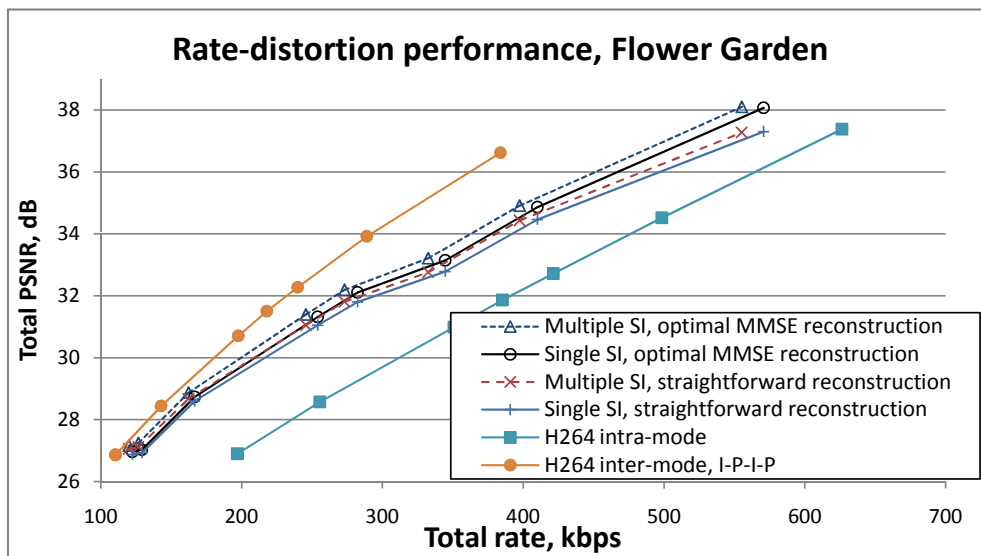
Under this assumption, the optimal minimum MSE reconstruction is given by the formula:

$$\hat{x}_{k,i}^{mmse, MH} = \frac{\sum_{j=1}^2 \int_l^u x p_{x|y_{k,i}^j}(x|y^j) dx}{\sum_{j=1}^2 \int_l^u p_{x|y_{k,i}^j}(x|y^j) dx} = \frac{\overbrace{\sum_{j=1}^2 \int_l^u \frac{\alpha_{k,i}}{2} x e^{-\alpha_{k,i}|x-y_{k,i}^j|} dx}^{\text{given by (4.39)}}}{\underbrace{\sum_{j=1}^2 \int_l^u \frac{\alpha_{k,i}}{2} e^{-\alpha_{k,i}|x-y_{k,i}^j|} dx}_{\text{given by (4.34)}}}, \quad (4.43)$$

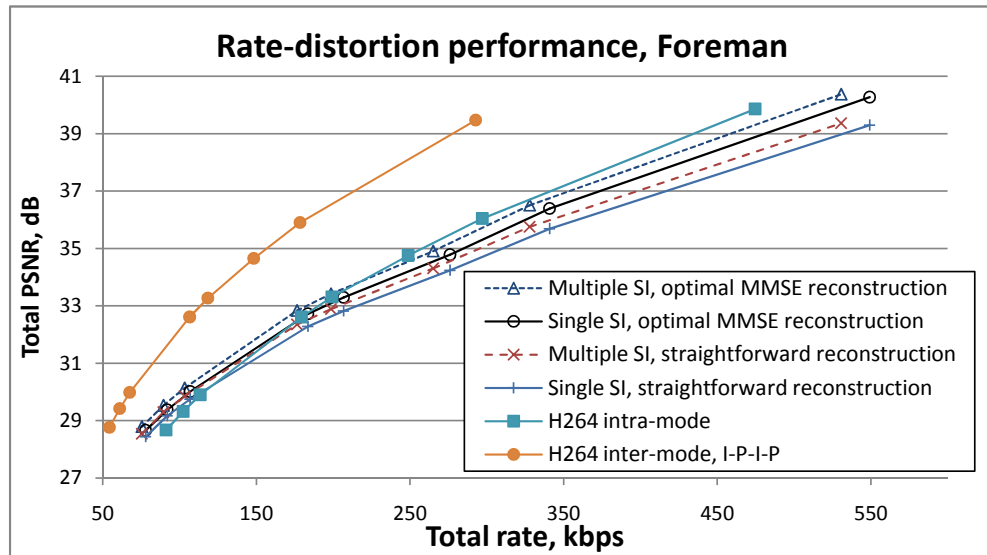
where all integrals can be calculated using closed-form expressions derived above. We don't derive here a single closed-form expression for (4.43) as we done it in (4.40), because it is cumbersome even for two side information (SI) hypotheses. On the other hand, the advantage of the form (4.43) is that it is easily extensible to the case of more than two SI hypotheses.



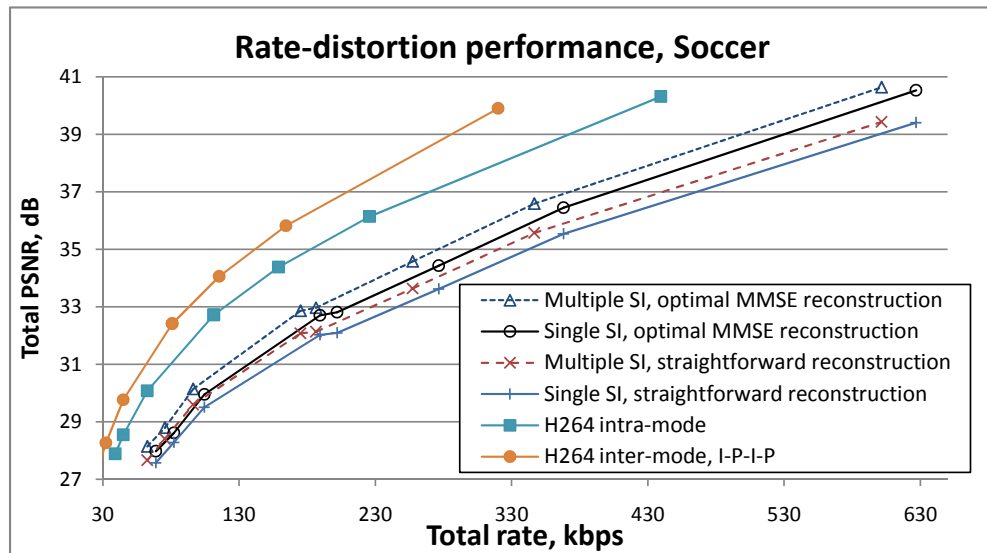
(a) Coastguard sequence, QCIF/15 Hz, 149 frames



(b) Flower Garden sequence, QCIF/15 Hz, 125 frames



(c) Foreman sequence, QCIF/15 Hz, 149 frames



(d) Soccer sequence, QCIF/15 Hz, 149 frames

Figure 4.8: Rate-distortion performance with optimal MMSE reconstruction

4.3.3 Experimental results

The rate-distortion performance of the proposed optimal minimum MSE (MMSE) reconstruction method has been assessed using the DISCOVER codec, on several sequences at QCIF/15 Hz, with a GOP size of 2. Figure 4.8 shows the average PSNR of the Luminance component for both key and Wyner-Ziv frames versus the total bit-rate. We compare the optimal method with the straightforward reconstruction approach (4.27). In the multiple side information scenario, $y_{k,i}$ used for the straightforward reconstruction according to (4.27), is the average of the two side information hypotheses $y_{k,i}^1$ and $y_{k,i}^2$. The two hypotheses $y_{k,i}^1$ and $y_{k,i}^2$ are the mesh-based and the block-based side information, as described in Section 3.3. In this simulation, we are using the local Laplacian correlation model with parameters obtained as described by (4.6), and we do not apply the parameter updating approach described in Section 4.2.

As the reconstruction does not affect the bit-rate, only the PSNR is improved when using the optimal reconstruction. It can be observed for both single and multiple side information scenarios that the proposed method gains up to 1dB in PSNR compared to the straightforward reconstruction method. Note that using multiple side information improves the rate-distortion performance by up to 0.3dB compared to the case when only one side information is used. We have also plotted the performance of a conventional codec (H.264) both in intra and inter modes for comparison. As for the Wyner-Ziv codec, only the Luminance component was taken into account.

The additional complexity incurred at the decoder by using the optimal reconstruction is negligible in comparison to the complexity of other decoder modules. As it can be seen from Table 4.1, using the optimal reconstruction increases the decoding time (single side information hypothesis) only by 0.01 seconds per frame (median value), which constitutes only 0.01%-0.1% of the total decoding time per frame. Here a median is taken instead of an average to better handle outliers of the time measurement resulting from casual operating system activity.

4.4 Hybrid encoder/decoder rate control solution

In this section we present a hybrid encoder-decoder rate control solution. The general idea of this type of solution is to let the decoder still exercise the precise rate-control (by verifying the bit error rate of the decoded sequence), while charging the encoder with the task of coarse rate estimation. The coarse rate estimation is used as the initial rate sent by the encoder, and the decoder needs to request for more data only if it is not able to decode the sequence with this initial rate. This approach allows reducing the number of decoder requests for parity bits or syndromes, as well as the usage of the return channel.

It is evident that if the estimated coarse rate sent as the initial rate is an over-estimation of the minimal sufficient rate, then the rate-distortion performance of the

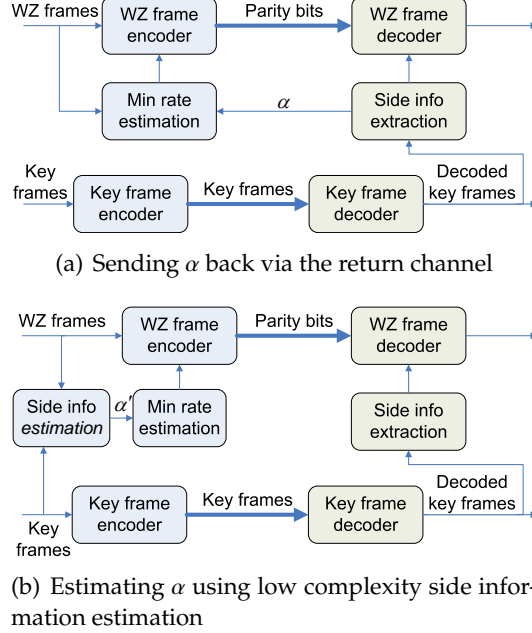
Table 4.1: Average decoding time per frame with single side information

Sequence	RD #	Simple Recons.	MMSE Recons.	Δt , sec
Coastguard	1	9.94s	9.95s	0.01
	2	11.44s	11.42s	-0.02
	3	13.48s	13.50s	0.02
	4	23.05s	23.07s	0.02
	5	24.98s	25.06s	0.08
	6	35.26s	35.31s	0.05
	7	44.75s	44.76s	0.01
	8	73.98s	74.00s	0.02
Flower Garden	1	11.02s	11.01s	0.00
	2	12.62s	12.62s	0.00
	3	15.39s	15.41s	0.02
	4	26.16s	26.20s	0.03
	5	29.42s	29.46s	0.04
	6	41.04s	41.20s	0.16
	7	47.32s	47.35s	0.03
	8	68.71s	68.70s	-0.01
Foreman	1	13.69s	13.49s	-0.19
	2	15.64s	15.65s	0.01
	3	18.32s	18.36s	0.04
	4	30.32s	30.32s	0.00
	5	34.02s	34.06s	0.03
	6	45.95s	46.02s	0.07
	7	56.16s	56.18s	0.02
	8	86.45s	86.22s	-0.24
Soccer	1	18.74s	18.73s	0.00
	2	21.13s	21.14s	0.01
	3	25.11s	24.73s	-0.38
	4	40.89s	40.66s	-0.23
	5	43.86s	43.88s	0.02
	6	57.02s	56.96s	-0.07
	7	70.74s	70.78s	0.04
	8	107.61s	107.51s	-0.09
		Median:		0.01

decoder is degraded. Therefore, the goal is to design a minimal rate estimator with a probability of under-estimating much higher than that one of over-estimating.

The correlation model parameter α required for the estimation can be obtained by the encoder in several ways, see Figure 4.9 for example. In the approach 4.9(a) α is estimated using the real side information, but without access to the original frame. The opposite approach 4.9(b) benefits from having access to the original data, but the real side information is usually unavailable (due to the limited encoder complexity). For the hybrid rate control we have adopted the first solution shown on Figure 4.9(a), as the precision of the decoder side algorithm developed by Brites *et al.* [BP08] is sufficient in this case.

An algorithm for minimal rate estimation using the WZ rate-distortion bound has been proposed by K. Lajnef [Laj06] (see Section 2.3.4.4). However, in practice this algorithm turns out to be inefficient in the sense that the estimated rate is very low, thus not allowing for a significant reduction of the number of decoder requests.

Figure 4.9: Possible ways of obtaining α at the encoder side

Below we present our algorithm, which is based on the Slepian-Wolf bound and thus better suited for estimating the rate for a single bit plane. After that we present the decoder complement of the hybrid rate control, which is an algorithm for a robust bit error rate estimation developed jointly with K. Lajnef. These results have been published in [KLG07].

4.4.1 Estimation of the minimal theoretical rate

Let \mathbf{x}_k^b , $1 \leq b \leq M_k$, represent the b -th bit plane of the band \mathbf{x}_k , $k = 1, \dots, 16$ (for a 4x4 DCT-like transform), where 2^{M_k} is the number of quantisation levels for the band k . We assume a virtual binary symmetric channel (BSC) between the bit plane \mathbf{x}_k^b and the corresponding side information bit plane \mathbf{y}_k^b . We note however that there is a difference between what we understand under the two bit planes. While the source bit plane \mathbf{x}_k^b is obtained by quantising source symbols (see Section 2.3.4.2), the side information bit plane \mathbf{y}_k^b is an estimation of the corresponding source bits using the side information vector \mathbf{y}_k along with previously decoded source bit planes \mathbf{x}_k^c , $c < b$. In other words:

$$y_{k,i}^b = \arg \max_{i=0,1} \left[\Pr(x_{k,i}^b = t | y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1}) \right], \quad (4.44)$$

where $\Pr(x_{k,i}^b = t | y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1})$ is computed according to the symbol-level correlation model used. For a Laplacian correlation model with conditional p.d.f. $p_{x|y}(x|y_{k,i}) =$

$\frac{\alpha_{k,i}}{2}e^{-\alpha_{k,i}|x-y_{k,i}|}$ this probability is given by:

$$\Pr(x_{k,i}^b = t | y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1}) = \int_{I^b(x_{k,i}^1, \dots, x_{k,i}^{b-1}, t)} \frac{\alpha_{k,i}}{2} e^{-\alpha_{k,i}|x-y_{k,i}|} dx, \quad (4.45)$$

where $I^b(x_{k,i}^1, \dots, x_{k,i}^{b-1}, t)$ denotes as in Section 4.2 the quantisation interval defined by b most significant bits (MSB) $(x_{k,i}^1, \dots, x_{k,i}^{b-1}, t)$ of the quantisation index.

Please note that we are using the notation $\alpha_{k,i}$ for consistency with the indexing of $x_{k,i}$ and $y_{k,i}$. In a real system when the solution shown on Figure 4.9(a) is used, only the global correlation model parameter α_k (one per band) is transmitted via the return channel (to keep the feedback rate low), thus $\alpha_{k,i} = \alpha_k \forall i$.

The minimal rate in the case of the distributed source coding with side information according to the Slepian-Wolf theorem is equal to the conditional entropy $H(X|Y)$ of X knowing Y (see Section 1.4.1). For a BSC the conditional entropy is equal to the entropy of the crossover probability:

$$R_{min}^b = H(X|Y) = H(p_{cr}) = -p_{cr} \log_2(p_{cr}) - (1 - p_{cr}) \log_2(1 - p_{cr}), \quad (4.46)$$

where $p_{cr} = \Pr(X \neq Y)$ (see Section 1.3.1).

A crossover event happens if:

$$x_{k,i}^b \neq y_{k,i}^b, \quad (4.47)$$

which means that the decoder *a priori* estimation of the bit $x_{k,i}^b$ is false. The condition (4.47) cannot however be calculated in practice because it requires the knowledge of both $y_{k,i}$ and $x_{k,i}$. In fact, the encoder only has $x_{k,i}$, and the correlation model parameter α (using one of the techniques shown on Figure 4.9 for example). Using this information, the encoder can calculate the *expectation* of $\Pr(x_{k,i}^b = t | y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1})$ (denoted below by $\Pr(t | y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1})$ for brevity):

$$\mathbb{E}_{p_{y|x_{k,i}}} [\Pr(t | y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1})] = \int_{-\infty}^{+\infty} \left[\int_{I^b(x_{k,i}^1, \dots, x_{k,i}^{b-1}, t)} \frac{\alpha_{k,i}}{2} e^{-\alpha_{k,i}|x-y_{k,i}|} dx \right] \underbrace{\frac{\alpha_{k,i}}{2} e^{-\alpha_{k,i}|y-x_{k,i}|}}_{p_{y|x_{k,i}}(y, x_{k,i})} dy. \quad (4.48)$$

If $I^b(x_{k,i}^1, \dots, x_{k,i}^{b-1}, t)$ is a continuous interval $[l, u]$ (this is always true for any b and $(x_{k,i}^1, \dots, x_{k,i}^{b-1}, t)$ with the quantisation index mapping adopted in DISCOVER), then (4.48) can be simplified using (4.34) to:

$$\begin{aligned} \mathbb{E}_{p_{y|x_{k,i}}} [\Pr(t | y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1})] &= \int_{-\infty}^l \frac{e^{-\alpha_{k,i}(l-y)} - e^{-\alpha_{k,i}(u-y)}}{2} \frac{\alpha_{k,i}}{2} e^{-\alpha_{k,i}(x_{k,i}-y)} dy \\ &+ \int_l^u \left(1 - \frac{e^{-\alpha_{k,i}(y-l)} + e^{-\alpha_{k,i}(u-y)}}{2} \right) \frac{\alpha_{k,i}}{2} e^{-\alpha_{k,i}|y-x_{k,i}|} dy \\ &+ \int_u^{+\infty} \frac{e^{-\alpha_{k,i}(y-u)} - e^{-\alpha_{k,i}(y-l)}}{2} \frac{\alpha_{k,i}}{2} e^{-\alpha_{k,i}(y-x_{k,i})} dy. \end{aligned} \quad (4.49)$$

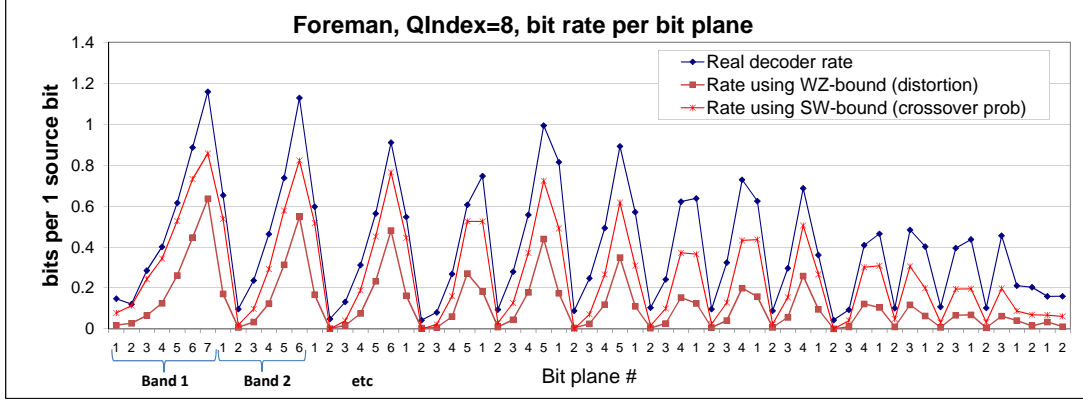


Figure 4.10: Estimated minimal rate vs. real decoder rate per bit plane

Using (4.34) further on and cancelling like terms, the expression (4.49) is finally reduced to:

$$\mathbb{E}_{p_{y|x_{k,i}}} \left[\Pr(t|y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1}) \right] = 1 - \frac{\left(1 + \frac{\alpha}{2}(x_{k,i} - l)\right) e^{-\alpha(x_{k,i} - l)} + \left(1 + \frac{\alpha}{2}(u - x_{k,i})\right) e^{-\alpha(u - x_{k,i})}}{2}. \quad (4.50)$$

The crossover probability p_{cr} for the whole bit plane b is estimated as:

$$\hat{p}_{cr} = \frac{1}{N} \sum_i I_{cr}(x_{k,i}), \quad (4.51)$$

where N is the size of the bit plane, and $I_{cr}(x_{k,i})$ is the indicator function of the crossover event:

$$I_{cr}(x_{k,i}) = \begin{cases} 1, & \text{if } x_{k,i}^b \neq \arg \max_{t=0,1} \mathbb{E}_{p_{y|x_{k,i}}} \left[\Pr(x_{k,i}^b = t|y_{k,i}, x_{k,i}^1, \dots, x_{k,i}^{b-1}) \right], \\ 0, & \text{otherwise.} \end{cases} \quad (4.52)$$

The minimal theoretical rate R_{min}^b for the bit plane b is then estimated as follows:

$$\hat{R}_{min}^b = -\hat{p}_{cr} \log_2(\hat{p}_{cr}) - (1 - \hat{p}_{cr}) \log_2(1 - \hat{p}_{cr}). \quad (4.53)$$

The described algorithm is compared to the approach of Lajnef [Laj06] on Figure 4.10. It is clear that the proposed algorithm permits to reduce the number of decoder requests more efficiently as its estimation is closer to the real decoder rate, which is confirmed by tests on full sequences: Figure 4.11 shows the average number of requests per frame (including all bit planes of all bands) for “Foreman” and “Hall monitor” sequences. It can be seen that the proposed method (marked as “With Rmin (SW bound)”) permits to reduce this number by up to the factor of 2 with respect to the approach of Lajnef [Laj06] (marked as “With Rmin (WZ bound)”), and by up to the factor of 4 with respect to the original DISCOVER codec without minimal rate estimation. Rate-distortion performance results showing only a slight deterioration (due to over-estimations of the minimal sufficient rate) with respect to fully decoder rate control, are presented in Section 4.4.3 below.

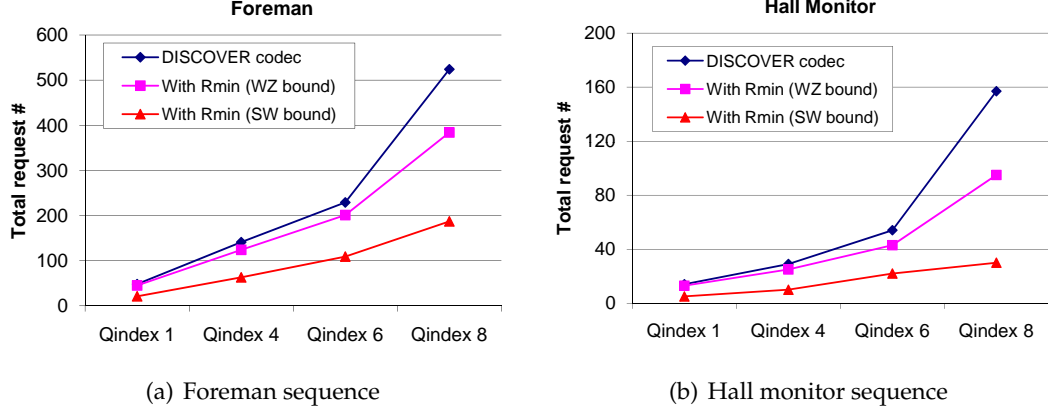


Figure 4.11: Average number of decoder requests per frame

4.4.2 Robust bit error rate estimation

First implementations of Stanford's Wyner-Ziv video coding architecture were using "ideal" error detection, i.e. the decoded bit plane has been compared with the original bit plane (as described in Section 2.3.2.2), which is absolutely unrealistic. Thus the problem of designing a realistic error rate measure has emerged. In this section we present a method for robust bit error rate estimation at the output of the turbo decoder. It consists of the two criteria: the confidence measure, and the check sum.

The confidence measure for a single bit at the output of the turbo decoder is derived from its log-likelihood ratio (LLR) Λ_i :

$$\Lambda_i = \log \frac{\Pr(x_{k,i}^b = 1)}{\Pr(x_{k,i}^b = 0)}, \quad (4.54)$$

where $\Pr(x_{k,i}^b)$ is the *a posteriori* probability of the bit $x_{k,i}^b$. The absolute value $|\Lambda_i|$ of the LLR is compared with a predefined threshold corresponding to the desired level of certainty. For example, requiring that

$$\max(\Pr(x_{k,i}^b = 1), \Pr(x_{k,i}^b = 0)) \geq 0.99 \quad (4.55)$$

amounts to set the threshold to $T = \log 99 \approx 4.5951$. If $|\Lambda_i| > T$, then the bit $x_{k,i}^b$ is considered certain. The confidence measure of the whole bit plane is calculated as the ratio of certain bits:

$$ConfScore = \frac{1}{N} \sum_i I_{|\Lambda_i| > T}, \quad (4.56)$$

where $I_{|\Lambda_i| > T}$ is the indicator function of the event $|\Lambda_i| > T$, and N is the size of the bit plane.

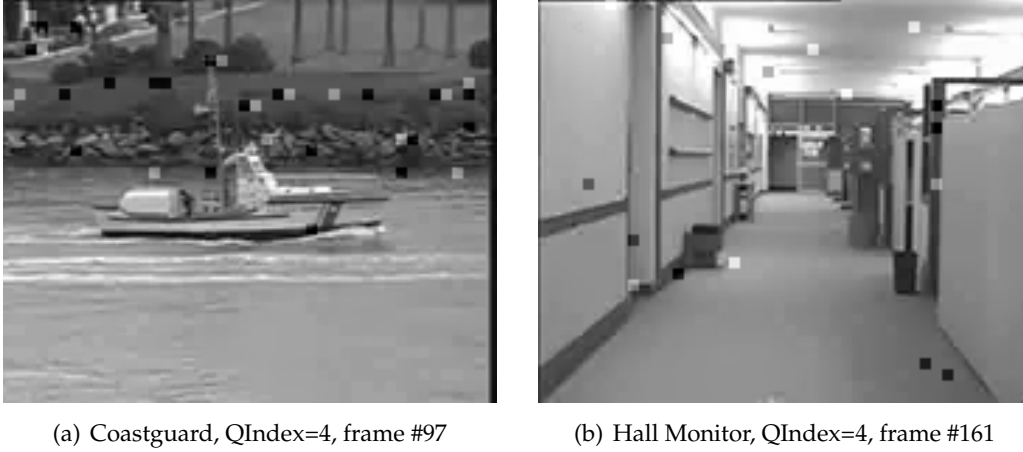


Figure 4.12: Artifacts due to the wrong estimation of the bit plane residual error rate

The original approach by Lajnef [Laj06] consists in using this confidence measure alone, declaring a bit plane to be successfully decoded if its confidence score is above $(1 - 10^{-3})$.

In practice, however, this criterion is not sufficient for robust error detection, as LLRs at the output of the turbo decoder are not always correct (due to the fact that the correlation model used for estimating *a priori* channel probabilities is not known perfectly). This results in a non-zero probability of having bit planes in the decoded stream with residual error rate exceeding 10^{-3} , which leads to wrong reconstruction of respective DCT coefficients. These artifacts are clearly visible in the decoded video sequence, see Figure 4.12 for example.

To enhance the robustness of the residual error rate estimation we employ a cyclic redundancy checksum (CRC). This checksum is calculated by the encoder for each bit plane, and transmitted to the decoder. The CRC is computed as the remainder of the division of the polynomial, corresponding to a sequence of bits (a bit plane in our case), by another polynomial chosen beforehand. A description of an algorithm computing CRC can be found in [Wil].

It is worth to note that a CRC of a sufficient strength can be used alone to detect the errors in the decoded bit stream with high probability. However, transmitting a “strong” CRC would penalise the rate-distortion performance of the codec by increasing the rate significantly. On the other hand, the above approach with the confidence measure costs nothing in terms of rate-distortion performance, thus it would be beneficial to couple error detection capabilities of the two methods, while keeping the size of the CRC reasonably small.

This is achieved by using an 8-bit CRC (for QCIF sequences with the size of a bit plane equal to 1584), and performing the check only for those bit planes that have passed the confidence measure control. If the CRC does not match the value received

from the encoder, then more parity bits are requested. Our experiments presented below show that this approach drastically reduces the probability of having artifacts like shown on Figure 4.12, without any visible impact on the rate-distortion performance.

4.4.3 Experimental results

The rate-distortion performance of the proposed method has been assessed with the DISCOVER codec on several video sequences with a GOP size of 2. The confidence threshold has been set to $T = 4.6$. An 8-bit CRC has been computed with the standard polynomial $x^8 + x^2 + x + 1$ for each bit plane.

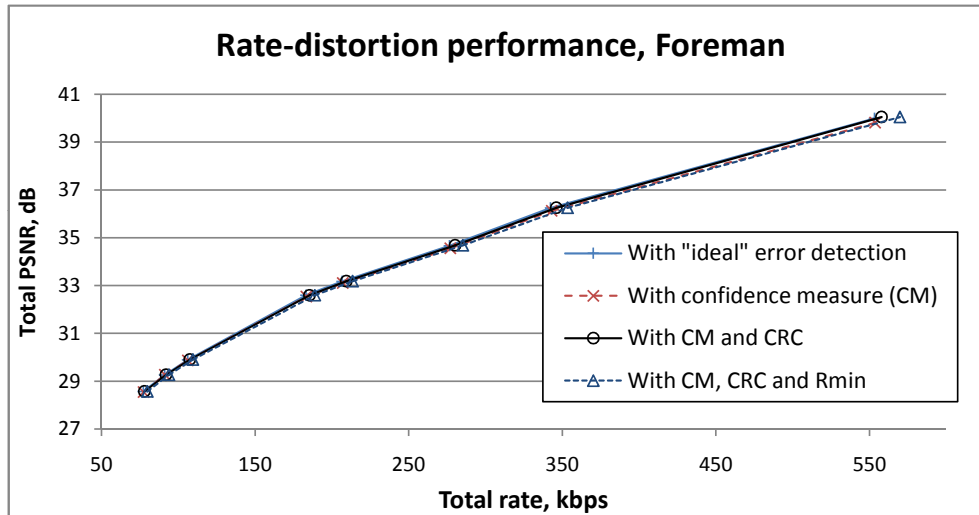
Figure 4.13 shows the average PSNR of the Luminance component for both key frames and Wyner-Ziv frames versus the total bit rate. It can be observed that, when using the proposed hybrid encoder/decoder rate control solution, there is almost no rate-distortion deterioration compared to the "ideal" error detection. A small rate augmentation in the case when minimum rate estimation is used, results from rare over-estimations of the minimal sufficient rate. These over-estimations can be explained by the fact that the exploited model for the correlation noise does not fit perfectly the real data. The visual quality is significantly improved, the annoying local artifacts are removed, as shown on sample frames in Table 4.2.

4.5 Rate-distortion optimisation

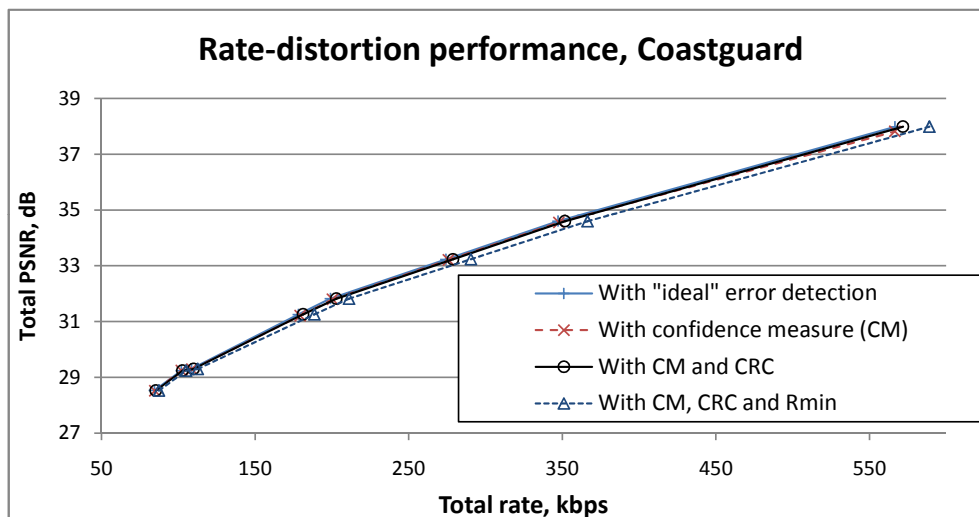
The DISCOVER codec uses constant quantisation tables for the whole video sequence. More precisely, for each rate-distortion (RD) point and for each video sequence a pair of two values (QIndex, QP) is defined off-line (i.e. before actual encoding), so that the *average* quality of Wyner-Ziv (WZ) decoded frames (encoded with quantisation table defined by QIndex) was close to the *average* quality of key frames (encoded with quantisation parameter QP). This approach suffers from several limitations, namely:

- Correspondence QIndex-QP should be established by the encoder beforehand. This means that the encoder should be able to perform decoding of WZ frames (probably several times), which is not always possible if the encoder is limited in complexity;
- Motion activity (and thus the quality of the side information) may vary at different time instants in the video sequence, thus the constraint on the *average* quality of WZ frames (to be close to the key frames quality) does not imply that the same condition is verified locally for adjacent frames (which is more important for subjective quality assessment than the average quality).

In this section we propose an algorithm which solves the two above problems by defining a quantisation table for each WZ frame independently, and in such a manner



(a) Foreman sequence, QCIF/15 Hz, 149 frames



(b) Coastguard sequence, QCIF/15 Hz, 149 frames

Figure 4.13: Rate-distortion performance with confidence measure and CRC

Table 4.2: Sample decoded frames with and without CRC











	Without CRC	With CRC
Coastguard, QIndex 4, frame 97		
FlowerGarden, QIndex 6, frame 5		
Foreman, QIndex 8, frame 21		
Hall Monitor, QIndex 4, frame 161		
Soccer, QIndex 1, frame 75		

Table 4.3: Scale coefficients c_k of the DISCOVER DCT-like integer transform

0.625	0.6423	0.625	0.6423
0.6423	0.625	0.6423	0.625
0.625	0.6423	0.625	0.6423
0.6423	0.625	0.6423	0.625

that the quality of the WZ frame is close to the quality of adjacent key frames, and the rate is as low as possible. This requires the knowledge of the actual correlation parameter (α), which can be obtained using either of the two solutions shown on Figure 4.9.

4.5.1 Quantiser design

The goal of the rate-distortion optimisation of the quantisation table for WZ frames is to find a set of 16 nonnegative integer values $\mathbf{q} = \{q_1, \dots, q_{16}\}$ (for a 4x4 DCT) defining the number of quantisation levels per band, so that the distortion $D(\mathbf{q})$ of the reconstructed WZ frame was below a certain threshold D^0 , and the rate $R(\mathbf{q})$ was the minimal over all possible \mathbf{q} :

$$\mathbf{q}^{\text{opt}} = \arg \min_{\substack{\mathbf{q} \in \mathbf{Q} \\ D(\mathbf{q}) < D^0}} R(\mathbf{q}), \quad (4.57)$$

where $\mathbf{Q} = \{\mathbb{N} \cup \{0\}\}^{16}$ is the set of possible quantisation matrices for 16 DCT bands.

To avoid a minimisation over the whole domain \mathbf{Q} (which is huge), the minimisation is carried out per band using the following assumption (see Section 1.2.2.3):

$$R(\mathbf{q}) = \sum_{k=1}^{16} R_k(q_k), \quad (4.58)$$

$$D(\mathbf{q}) \approx \sum_{k=1}^{16} D_k(q_k), \quad (4.59)$$

where $R_k(q_k)$ and $D_k(q_k)$ are the rate and distortion of k -th DCT band. While (4.59) is true for the DCT transform, the bands of the integer DCT-like transform used in DISCOVER represent corresponding DCT bands scaled by some coefficients c_k . These coefficients are listed in Table 4.3. Therefore to relate the distortion $D(\mathbf{q})$ in the pixel domain to distortions $D_k(q_k)$ per band, (4.59) needs to be transformed into:

$$D(\mathbf{q}) \approx \sum_{k=1}^{16} c_k^2 D_k(q_k). \quad (4.60)$$

The method of Lagrange multipliers is used to find an optimal allocation of the

rate and distortion between bands (see Section 1.2.2.3):

$$\mathbf{q}^{\text{opt}} = \arg \min_{\substack{\lambda \in \mathbb{R}_+ \\ \sum_k c_k^2 D_k(q_k) < D^0}} \min_{\mathbf{q} \in \mathbf{Q}} \left(\sum_{k=1}^{16} R_k(q_k) + \lambda \sum_{k=1}^{16} c_k^2 D_k(q_k) \right) \quad (4.61)$$

$$= \arg \min_{\substack{\lambda \in \mathbb{R}_+ \\ \sum_k c_k^2 D_k(q_k) < D^0}} \sum_{k=1}^{16} \min_{q_k \in \mathbb{N} \cup \{0\}} (R_k(q_k) + \lambda c_k^2 D_k(q_k)). \quad (4.62)$$

The minimisation of $(R_k(q_k) + \lambda c_k^2 D_k(q_k))$ is done via estimating the distortion-rate function $D_k(R_k)$ for the band k . Let us first describe the estimation algorithm, and then we will return to the optimisation problem (4.62).

4.5.2 Distortion-rate function estimation

The distortion-rate function $D_k(R_k)$ for the band k can be approximated by a convex envelope of a set of points $(D_k(q_j), R_k(q_j))$ for q_j belonging to a representative subset S_k of the set $Q_k = \mathbb{N} \cup \{0\}$ of all possible quantisers. In our experiments S_k includes all even integers from 0 to 128: $S_k = \{0, 2, \dots, 126, 128\}$.

Please note that we do not require that q_k was a power of two, which is a necessary condition for a RD-optimal quantiser in conventional predictive coding. In contrast to the conventional case, in Wyner-Ziv coding the rate also depends on the size of the quantisation interval, and not only on the minimal number of bits required for a quantisation index representation. Indeed, the correlation of the quantisation index with the side information is higher when the quantisation interval is larger, therefore the decoding of that quantisation index with the side information would require less parity bits. The reconstruction distortion is also higher with larger interval. Therefore the compromise between the quantisation interval size and the number of bits in the binary representation of a quantisation index should be found. As we consider only uniform (for the DC band) and dead-zone uniform (for all AC bands) quantisers, these two parameters are both defined by the number of quantisation levels, which is varied to find the best solution. Our experiments confirm that the quantiser q_k minimising $(R_k(q_k) + \lambda c_k^2 D_k(q_k))$ may have any integer number of quantisation levels, and not necessarily a number which is a power of two.

The rate $R_k(q_j)$ and distortion $D_k(q_j)$ for a given number of quantisation levels q_j need to be estimated. For the rate estimation, we employ the algorithm described in Section 4.4.1:

$$R_k(q_j) = \sum_{b=1}^{\lceil \log_2 q_j \rceil} \hat{R}_{\min}^b, \quad q_j > 0 \quad (4.63)$$

where \hat{R}_{\min}^b is given by (4.53) for a uniform or dead-zone quantiser with q_j levels. For $q_j = 0$ the rate $R_k(0)$ is zero.

The distortion $D_k(q_j)$ is computed using the approach described in Section 2.3.4.4. First, an appropriate quantiser (uniform or dead-zone uniform) with q_j levels is created. After that, assuming the straightforward decoder reconstruction (4.27), the expected distortion of a coefficient $x_{k,i}$ is given by (2.51) with $b = \lceil \log_2 q_j \rceil$ (when all $\lceil \log_2 q_j \rceil$ bit planes are decoded):

$$\hat{D}(x_{k,i}) = \frac{2 - [1 + \alpha_{k,i}(x_{k,i} - l)] e^{-\alpha_{k,i}(x_{k,i} - l)} - [1 + \alpha_{k,i}(u - x_{k,i})] e^{-\alpha_{k,i}(u - x_{k,i})}}{\alpha_{k,i}^2}, \quad (4.64)$$

where $[l, u]$ is the quantisation interval for $x_{k,i}$, and $\alpha_{k,i}$ is the respective correlation model parameter. The expected distortion for the whole band is then given by:

$$D_k(q_j) = \frac{1}{N} \sum_i D(x_{k,i}), \quad (4.65)$$

where N is the size of the band.

When the estimated rate $R_k(q_j)$ and distortion $D_k(q_j)$ are computed for all $q_j \in S_k$, the convex envelope of the set of points $(D_k(q_j), R_k(q_j))$ in space $R \times D$ is constructed. This convex envelope is taken as an approximation of the distortion-rate function $D_k(R_k)$ for the band k .

4.5.3 Optimisation

Given the 16 distortion-rate functions $D_k(R_k)$, $k = 1, \dots, 16$, the minimisation of (4.62) is carried out as follows (see Section 1.2.2.3 for more details).

Minimising $(R_k(q_k) + \lambda c_k^2 D_k(q_k))$ is equivalent to finding a common slope (with tangent $-\lambda$) on all 16 nonincreasing distortion-rate functions $D_k(R_k)$. Therefore, the global minimisation in (4.62) consists in finding the values of λ , which satisfy the condition $\sum_k c_k^2 D_k(q_k) < D^0$ in points $q_k(\lambda)$ corresponding to the slope with tangent $-\lambda$, and then finding the minimum of the Lagrangian function $\sum_k (R_k(q_k) + \lambda c_k^2 D_k(q_k))$ for all those λ .

As proposed in Section 1.2.2.3, the search for λ is done using dichotomy. For implementation convenience the variable $\lambda \in \mathbb{R}_+$ is substituted with the variable $\lambda' = \frac{\lambda}{1+\lambda}$ having a limited definitional domain: $0 \leq \lambda' \leq 1$. Thus (4.62) is transformed into:

$$\mathbf{q}^{\text{opt}} = \arg \min_{\substack{0 \leq \lambda' \leq 1 \\ \sum_k c_k^2 D_k(q_k) < D^0}} \sum_{k=1}^{16} \min_{q_k \in \mathbb{N} \cup \{0\}} ((1 - \lambda') R_k(q_k) + \lambda' c_k^2 D_k(q_k)). \quad (4.66)$$

The starting interval $[0, 1]$ is recursively sub-divided into two equal intervals until a value of λ' is found satisfying $\sum_k c_k^2 D_k(q_k) < D^0$. After that, the value of λ' is varied around this found point to find the minimum of the Lagrangian function $\sum_k (R_k(q_k) + \lambda c_k^2 D_k(q_k))$. The set of q_k corresponding to the minimum of the Lagrangian function, is the RD-optimal quantisation table for the current WZ frame under the

given distortion constraint. In our experiments we have set D^0 so that the expected difference in PSNR between the WZ frame and adjacent key frames was less than 0.5 dB.

4.5.4 Experimental results

We have evaluated the proposed algorithm within the DISCOVER codec. We are keeping quantisation parameters for key frames the same, which allows us to compare with the original version of DISCOVER with predefined static correspondences between the quantisation parameter and the quantisation table index (usually denoted as QIndex) for each video sequence. Test video sequences have spatial resolution QCIF (176x144), have temporal resolution 15 Hz, and are monochrome (only Luminance is encoded). In contrast to previous simulations, here the correlation model parameter α is estimated at the encoder, and not transmitted by the decoder. The motivation for this, as well as the estimation method employed, are presented in the next section.

Rate-distortion performances are shown on Figure 4.14. With respect to the original DISCOVER codec, the overall rate-distortion performance is improved by up to 0.5 dB. However, the main improvement is shown by an example on Figure 4.15, depicting frame by frame PSNR of the decoded sequence. Indeed, the difference in quality between adjacent WZ and key frames is significantly decreased, from average 1.18 dB to 0.36 dB. The maximal difference of 4.28 dB without rate-distortion optimisation is reduced to 1.28 dB when the optimisation is used. Key frames in both cases have the same quality.

4.6 Robust estimation of correlation model parameter at the encoder

In this section we present an algorithm for robust correlation model parameter estimation at the encoder side (corresponding to the setup shown on Figure 4.9(b)). The motivation for developing such an algorithm is the sensitivity of the above rate-distortion optimisation method to the precision of the correlation model parameter. Indeed, as mentioned in Section 4.1, the method of Brites *et al.* [BP08] employed by the decoder to estimate the global correlation model (using motion compensated residual) often over-estimates the correlation. While this is not critical for the decoding (because of the presence of the return channel) or the minimal rate estimation (because it can only result in under-estimating the minimal rate), it might be dangerous for the rate-distortion optimisation of the WZ quantisation table. If the correlation is over-estimated, then the algorithm may decide to use too few quantisation levels, and the decoder will not be able to reconstruct the frame within the target quality constraint.

A typical relationship between the real correlation model parameter α_k^T and the decoder estimation $\hat{\alpha}_k$ based on the motion compensated residual, is shown on Figure

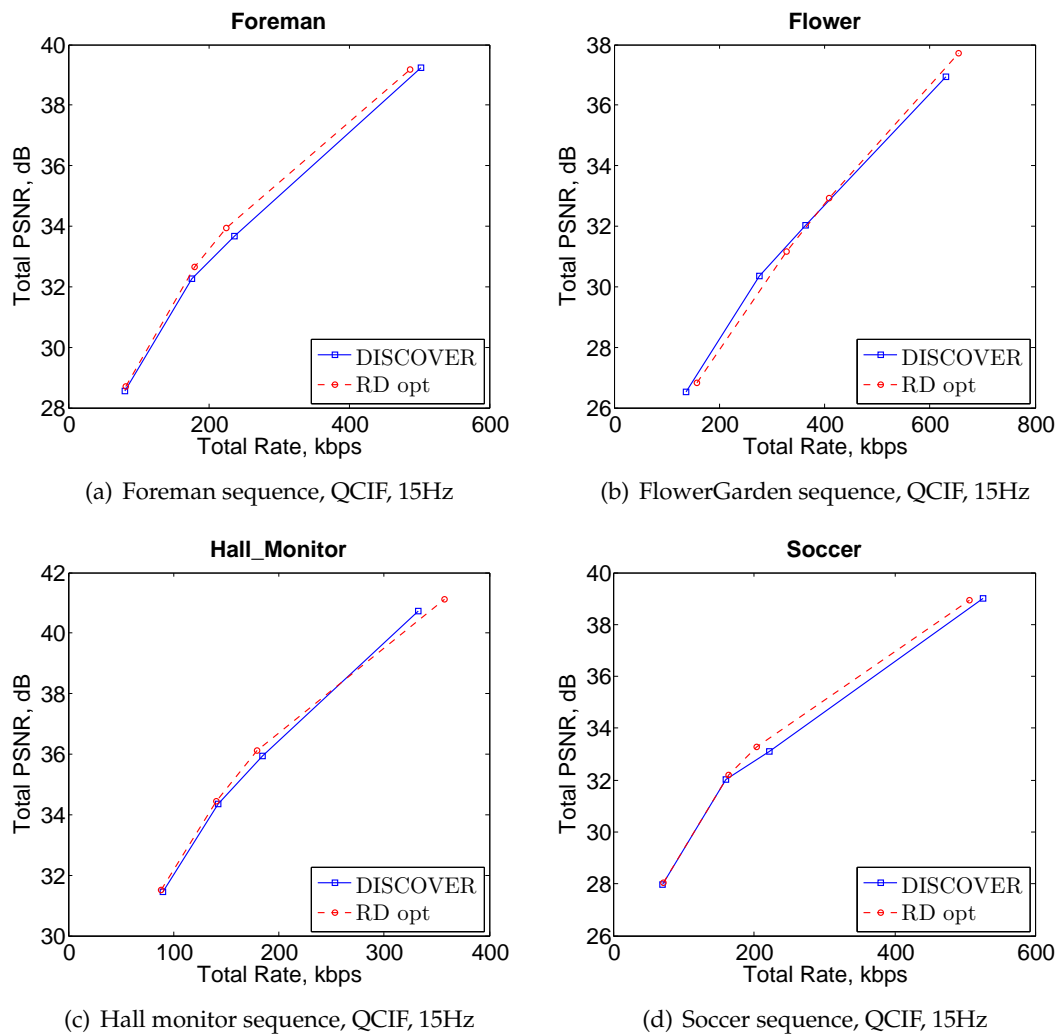


Figure 4.14: Rate-distortion performance with adaptive quantisation table for WZ frames

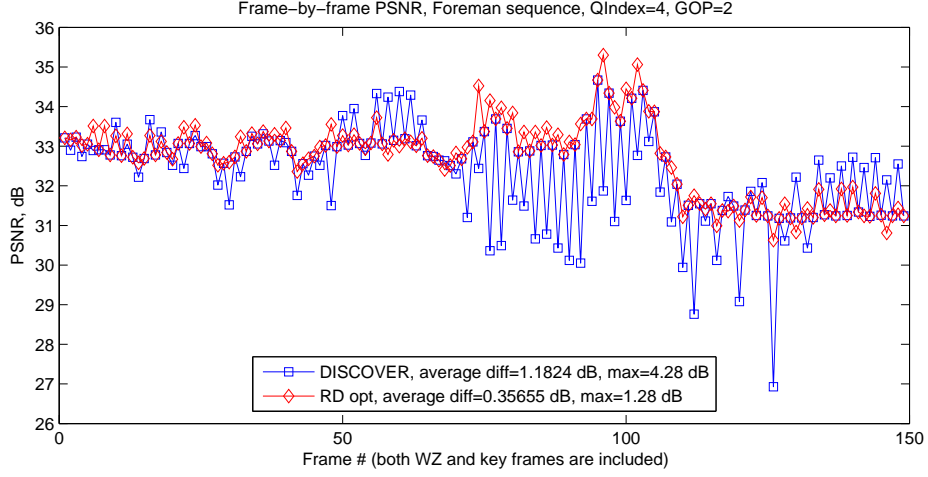


Figure 4.15: Frame by frame PSNR of the decoded sequence

4.16 for two DCT bands #1, #3 and #5. It is obvious that the parameter $\hat{\alpha}_k$ cannot be used for the correct distortion estimation of the decoded band (especially for high frequency bands).

It is also obvious that a robust estimation at the encoder would require performing the same side information extraction technique as used by the decoder. However, for having a good estimate of the global correlation model parameter α_k , one doesn't need to perform the side information extraction for the whole frame. In fact, a relatively small percentage of randomly selected blocks should allow for a good estimation.

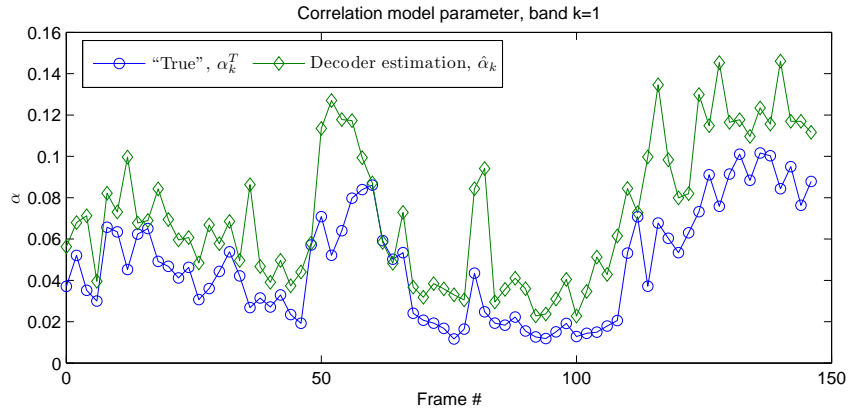
To determine a minimal sufficient percentage, we have plotted the evolution of the estimation while increasing the percentage of used samples for this estimation (see Figure 4.17). It can be seen that having the side information for 10% to 20% of all blocks of the frame is usually sufficient for an estimation which is much closer to the value α_k^T than the decoder estimation $\hat{\alpha}_k$.

The method is functioning as follows. First, a random block pattern is generated selecting 10% to 20% (an algorithm parameter) of all 16x16 blocks of the frame. Then the motion estimation is performed as described in Section 2.3.4.5, but only for those blocks which are selected in the pattern. The global correlation model parameter α_k^{est} is then estimated as described in Section 2.3.4.6 (taking into account only those blocks for which the side information has been created).

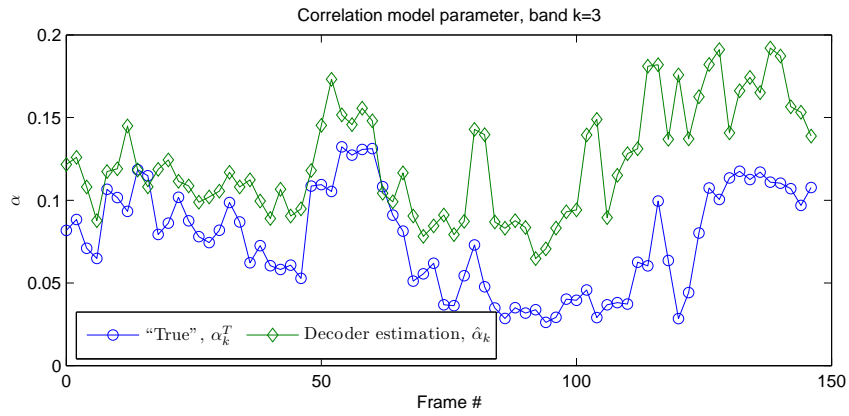
We have measured the overall estimation error of the proposed method for two sequences, Foreman and Soccer. This error is calculated as follows:

$$E = \sum_k MSE(\alpha_k^{est}, \alpha_k^T), \quad (4.67)$$

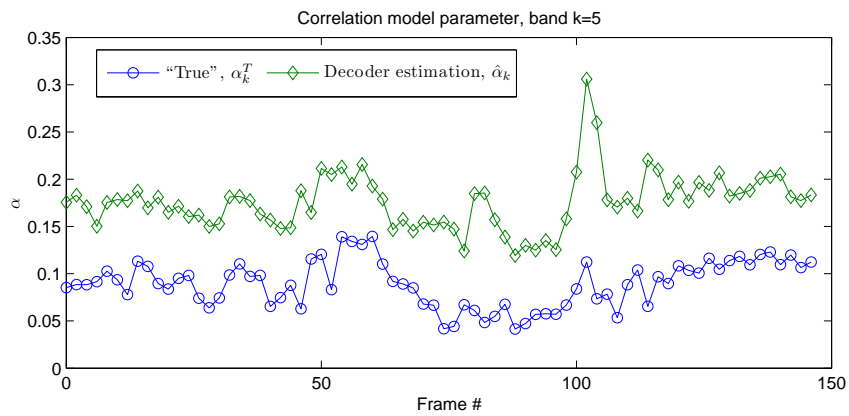
where MSE is the mean squared error taken over all frames in the sequence. This error E is plotted on Figure 4.18. The improved accuracy of the proposed method



(a) Band 1 (DC)

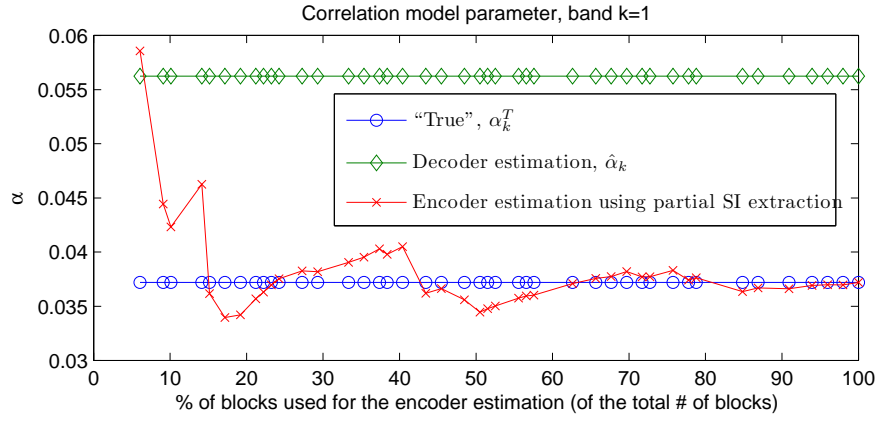


(b) Band 3

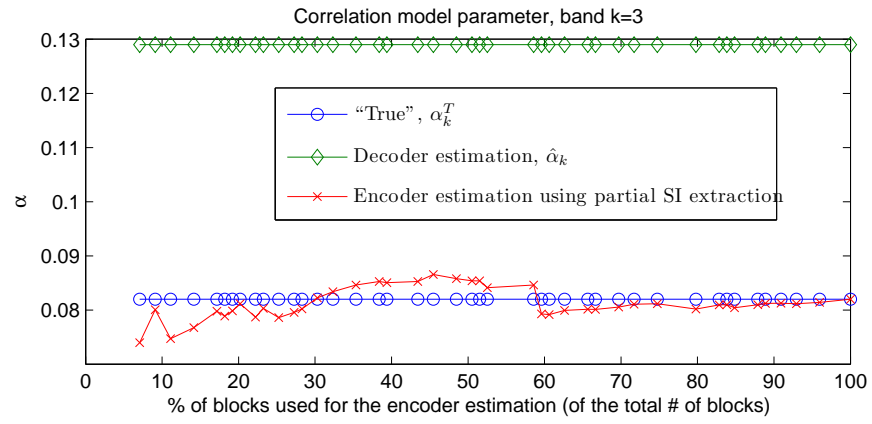


(c) Band 5

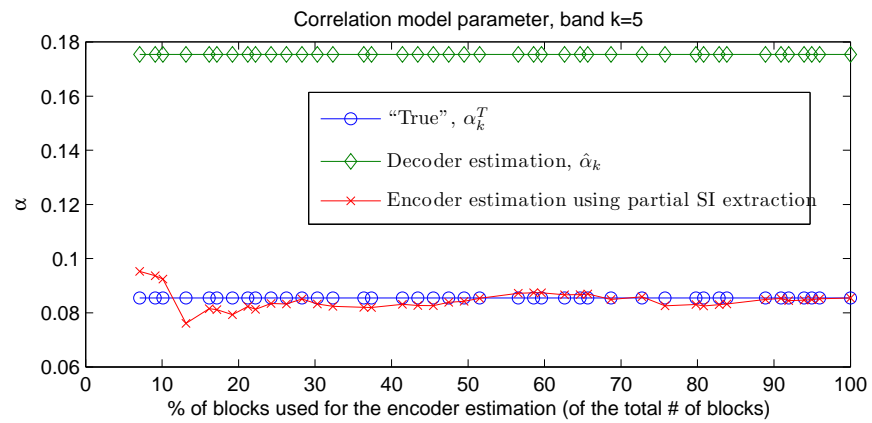
Figure 4.16: Global correlation model parameter, "true" vs. decoder estimation



(a) Band 1 (DC)



(b) Band 3



(c) Band 5

Figure 4.17: Model parameter estimation using partial SI extraction

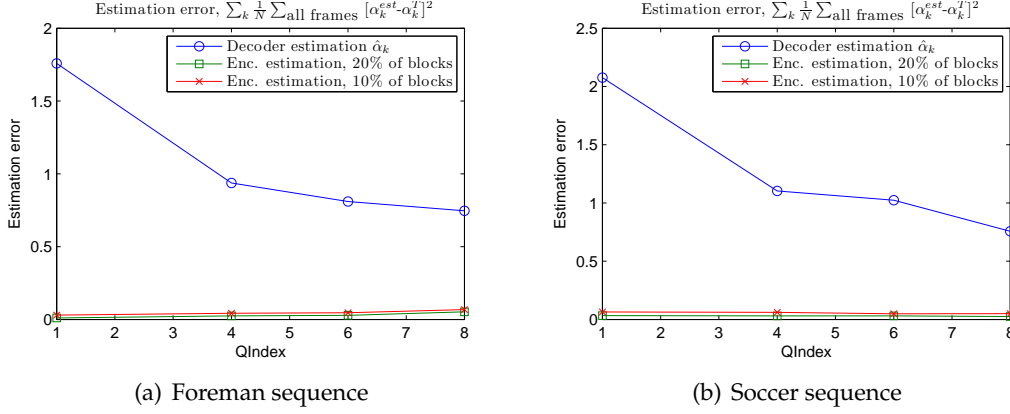


Figure 4.18: Correlation model parameter estimation error

with respect to the decoder estimation method by Brites *et al.* [BP08] is clearly visible.

We propose to use this estimation method instead of using the solution 4.9(a), when an accurate value of the correlation model parameter is required (as for the rate-distortion optimisation of the quantisation table, see Section 4.5). In spite of the fact that this method increases the encoder complexity, and breaks the assumption of the independent encoding of frames (the distributed source coding paradigm), it might sometimes be the only possible way of obtaining the actual correlation model between the source data and the side information at the encoder.

4.7 Mixed intra/WZ encoding mode

Another way of improving the rate-distortion performance of a Wyner-Ziv video codec based on the Stanford architecture [ARG04, GARRM05], is managing flexibly “intra” and Wyner-Ziv (WZ) encoding modes. In fact, in the original Stanford architecture there is no on-line “intra”/WZ mode decision module. Key frames (“intra” mode) and Wyner-Ziv frames are selected in a regular manner for the whole sequence, which is obviously sub-optimal, as video sequences are often non-uniform in terms of motion activity (which influences the correlation in the temporal domain and thus the efficiency of the WZ coding).

A possible solution consists in encoding the frames with intensive motion in “intra” mode, i.e. with a predictive h.264-like video encoder, which is very effective in exploiting the spatial correlation. The DISCOVER codec already includes such an algorithm proposed by J. Ascenso *et al.* [ABP06], which can be enabled at the encoder to allow an adaptive GOP size selection based on the locally estimated motion activity (see Section 2.3.4.1). The algorithm though has some inconveniences:

1. the job of adaptive GOP size selection is carried out by the encoder, thus requir-

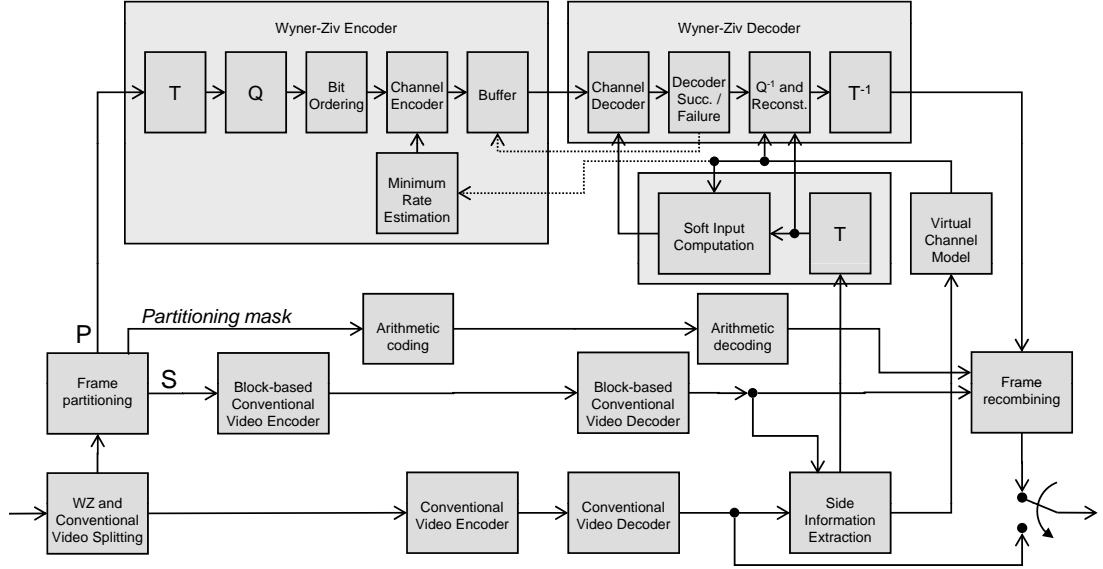


Figure 4.19: DISCOVER codec with mixed “intra” /WZ encoding mode decision

ing some extra computational power;

2. it requires a pre-processing of the video sequence (an extra processing pass), which introduces delay in the coding system, and requires the encoder to have more memory;
3. the decision is taken on the frame level that is sub-optimal; it would be nice to have a block level decision like in PRISM architecture [PR02, PR03b, PMR07] for example.

In this section we propose a solution which is similar to the approach used in PRISM. We partition the frame into two sub-frames, one of which is then encoded in “intra” mode, while the other is WZ-encoded. The pattern of the partitioning is determined on the encoder side using a simple motion detection technique, thus the complexity of the encoder is kept low.

4.7.1 An overview of the proposed approach

The DISCOVER codec architecture including the proposed approach is depicted on Figure 4.19. The Wyner-Ziv frame $I_{wz}(x, y)$ is partitioned into two subsets P and S using a certain block-wise pattern (the choice of the pattern is explained below). First, the partition mask is encoded using an arithmetic encoder. The subset S is then encoded block per block by a conventional h.264-like video encoder in “intra” mode. The blocks of the remaining subset P are encoded by WZ encoder normally, i.e. transformed, quantised, and encoded using a Slepian-Wolf encoder (the encoded bands

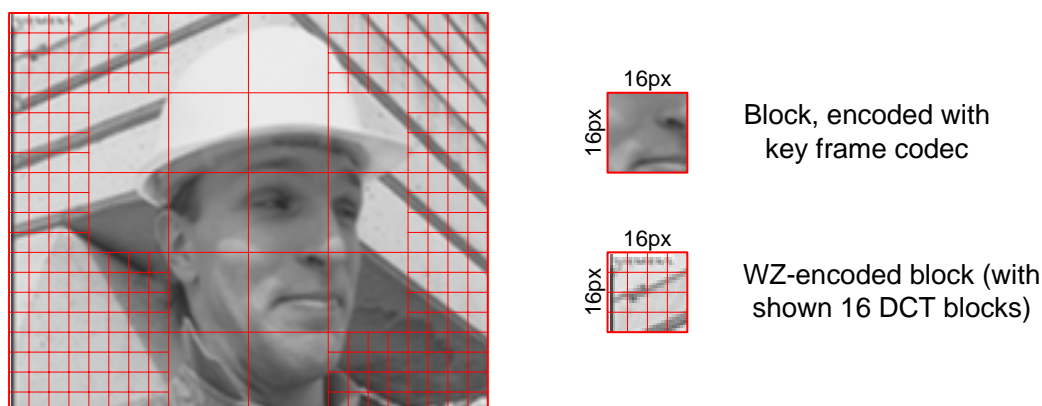


Figure 4.20: An example of frame partitioning

will contain less elements than normally, as the coefficients of the subset S need not to be encoded).

The WZ decoder first decodes the partition mask mandatory for determining the two subsets P and S . Then it decodes the subset S of the frame, encoded with the key frame codec, and arranges the blocks of S in their positions in the frame according to the mask. Now it is ready to decode the subset P in a regular manner, i.e. using the temporal side information generated using adjacent key frames. We are not using the subset S to construct the additional (“spatial”) side information for P as it is often done in pixel domain solutions, due to following reasons:

1. the correlation between adjacent DCT coefficients is much lower than the correlation between adjacent pixels (because the role of the DCT is to remove this correlation);
2. the construction of the pattern is done so that blocks of P are assumed to have good temporal side information, thus there is no need for a spatial SI.

4.7.2 The pattern of frame partitioning

The frame partitioning is done blockwise. The size of blocks was selected to match both the block structure of the 4x4 DCT and the smallest separate encoding unit of the used h.264 encoder implementation, which is the block of 16x16 pixels. So, the unit of the partition is a block of 16x16 pixels, which can either be encoded by the key frame encoder, or transformed using 4x4 DCT into 16 coefficients in each of 16 bands. An example partitioning is shown on Fig. 4.20.

The goal of frame partitioning is to separate blocks for which the temporal side information is good (and WZ coding is therefore effective), from the remaining part of the frame (which can be more effectively compressed using spatial prediction in

the key frame encoder). Given that the encoder computational power is limited, only simple techniques can be involved into this task.

The simplest method of motion detection consists in thresholding the frame difference between the two frames F_1 and F_2 :

$$FD_{F_1, F_2}(x, y) = F_2(x, y) - F_1(x, y). \quad (4.68)$$

If the absolute value of the frame difference $FD(x, y)$ is higher than a predefined motion detection threshold T_{MD} , then the pixel is declared "moving", otherwise it is considered "stationary":

$$Moving(x, y) = \begin{cases} 1 & \text{if } |FD_{F_1, F_2}(x, y)| \geq T_{MD}, \\ 0 & \text{otherwise.} \end{cases} \quad (4.69)$$

In the distributed video coding however we are interested in motion detection between frames $I_{ref}(x, y) - I_{wz}(x, y)$ and $I_{wz}(x, y) - I_{base}(x, y)$, as only pixels that are stationary on both intervals can be surely interpolated using frames $I_{ref}(x, y)$ and $I_{base}(x, y)$. Therefore we use the following motion detection criterion:

$$Moving_{DCT}(x, y) = \begin{cases} 0 & \text{if } (|FD_{I_{ref}, I_{wz}}(x, y)| \leq T_{MD} \ \& \ |FD_{I_{wz}, I_{base}}(x, y)| \leq T_{MD}), \\ 1 & \text{otherwise.} \end{cases} \quad (4.70)$$

The ratio of "moving" pixels per block 16x16 is then calculated:

$$MovRatio(u, v) = \frac{1}{256} \sum_{x, y=0}^{15} Moving_{DCT}(16u + x, 16v + y). \quad (4.71)$$

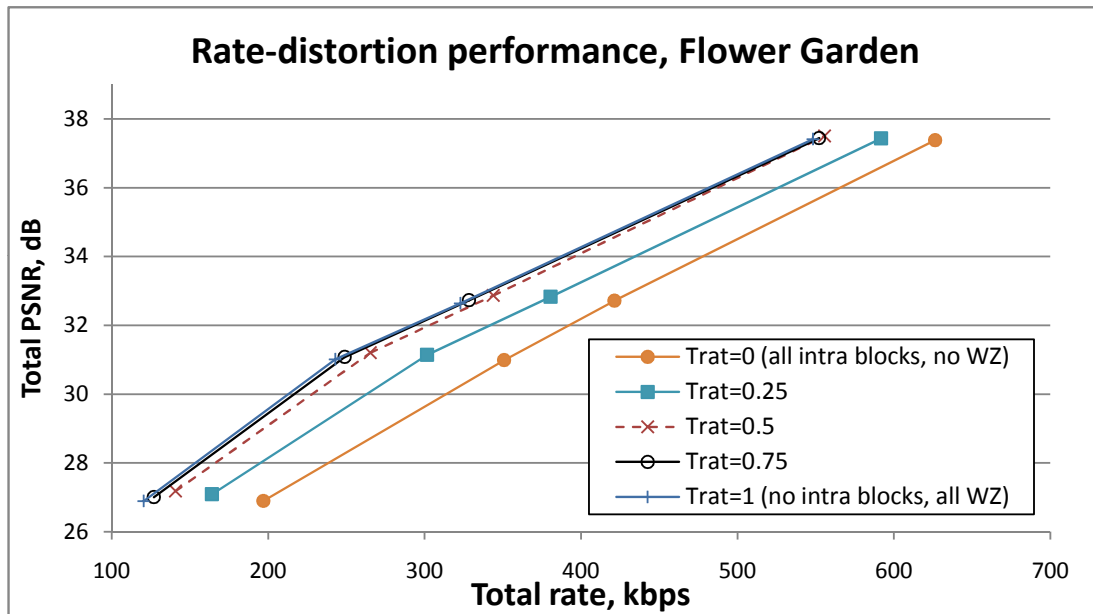
If this ratio exceeds another predefined threshold T_{rat} , then the block at position (u, v) is encoded with the key frame encoder ("intra"), otherwise it is WZ encoded:

$$\text{Block } (u, v) \text{ is encoded as: } \begin{cases} \text{intra} & \text{if } MovRatio(u, v) \geq T_{rat}, \\ \text{WZ} & \text{otherwise.} \end{cases} \quad (4.72)$$

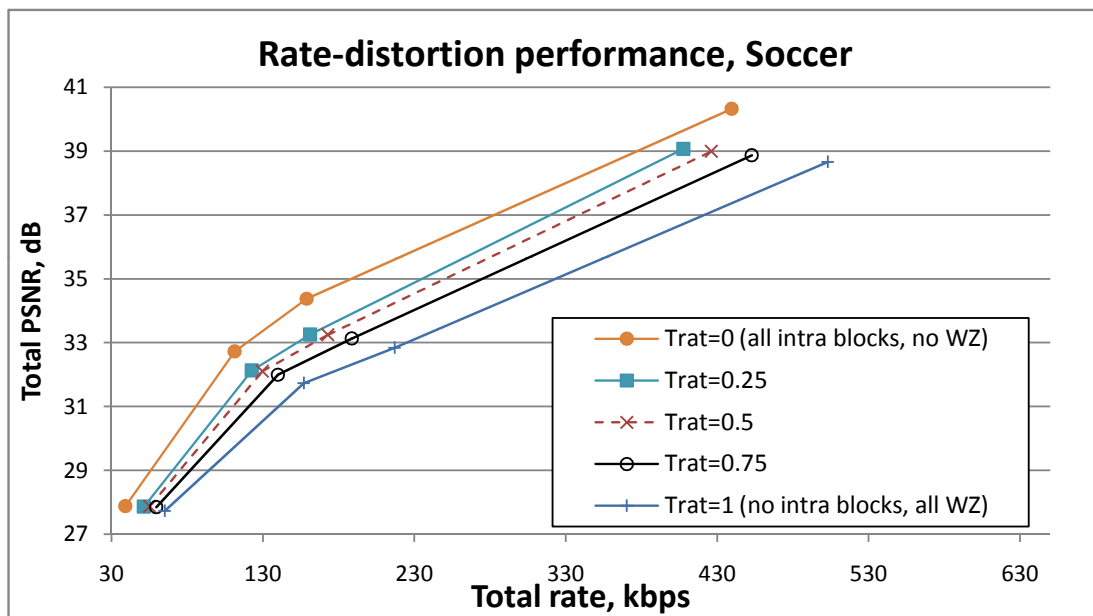
4.7.3 Threshold selection

The choice of thresholds T_{MD} and T_{rat} is very important for the algorithm. After some experiments, we have set the first one to 5, given that pixels' luminances are in the range $[0, 255]$.

The threshold T_{rat} influences the rate-distortion performance directly, so it has to be chosen with care. In order to find a good value, the response of the total encoding system to different values of T_{rat} was analysed. As part of the experiment, rate-distortion performances for two sequences with different motion content are shown on Figure 4.21.



(a) "Flower Garden" sequence



(b) "Soccer" sequence

Figure 4.21: Experiments for choosing T_{rat} threshold

The first sequence “Flower Garden” is characterised by slow camera panning. This type of motion can be easily estimated by the decoder, and thus the correlation of the original frame with the side information is high, and frames can be compressed more efficiently in the WZ mode than in “intra” mode. At the same time the frame difference here shows high motion activity, which causes the algorithm to encode many blocks in “intra” mode, reducing the overall performance. The graph 4.21(a) shows that T_{rat} should be kept as high as possible to not to allow the performance to drop significantly.

The other sequence called “Soccer” is characterised by very intensive and chaotic motion. For this sequence the “intra” encoding mode is by far more efficient than the WZ mode, and thus the smaller the permissive “intra” mode threshold T_{rat} , the better is the overall rate-distortion performance.

Looking at these two examples as extreme cases, and also looking at other sequences not mentioned here, we have set the threshold T_{rat} to the value 0.5, which ensures the good performance in a variety of test conditions.

$$\begin{cases} T_{MD} &= 5, \\ T_{rat} &= 0.5. \end{cases} \quad (4.73)$$

4.7.4 WZ frames mixed encoding

First of all, the partitioning mask should be encoded. As we have mentioned before, an arithmetic encoder is used to encode this mask at the rate close to its entropy. More precisely, we use the CABAC encoder of the h.264 standard [MSW03] with 5 contexts (one default and 4 others corresponding to all 4 possible configurations: 00, 01, 10, and 11, of the two mask bits preceding the current bit horizontally and vertically).

After that, all “intra” blocks are encoded by the key frame encoder. We use the same h.264 encoder which is used to encode key frames, with same parameters. The only difference is that it encodes individual blocks (possibly separated one from another), and not the whole frame at a time. That is why the encoding might be less efficient as the usage of spatial correlation is limited in this case.

The remaining part of the frame is then transformed, compacted in continuous bands (shrunk into bands of smaller size, filling up “holes” where “intra” blocks are), quantised, and encoded using a channel code. Parity bits (if turbo codes are used) or accumulated syndromes (if codes LDPC are used) are put into the buffer accessible for decoder requests.

4.7.5 WZ frames mixed decoding

The decoding is done in the same order. First the partitioning mask is decoded using the CABAC decoder. Then the block-based key frame decoder is used to reconstruct

the blocks encoded in “intra” mode. These blocks are put onto their places using the partitioning mask.

The side information also needs to be compacted using the partitioning mask in the same way as the WZ frame. The decoding then proceeds in the usual manner. Finally, the decoded parts of the WZ frame are combined together using the partitioning mask.

4.7.6 Experimental results

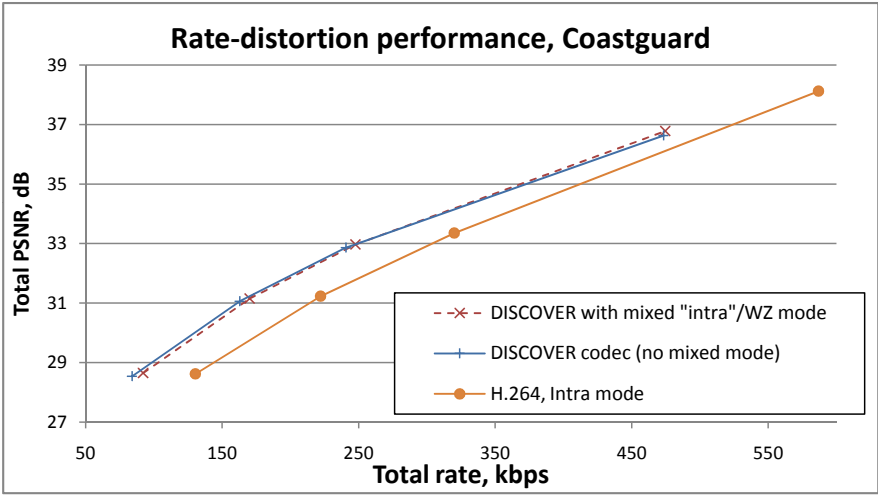
The proposed approach has been assessed using several QCIF video sequences having temporal resolution 15Hz, the results are shown on Figure 4.22. It has been observed that the rate-distortion performance improvement is significant (1-2 dB) for those sequences for which the “intra” mode of H.264 is preferable to the WZ coding in general (Foreman and Soccer sequences). For other sequences the gain is low, and sometimes even negative though small (FlowerGarden, for example). The proposed technique also helps reducing the decoding time for sequences with high motion activity, see Figure 4.23, due to the increased correlation for the WZ part of the frame. Therefore, we think that the described solution inspired by PRISM, should be used by any Wyner-Ziv video codec based on Stanford architecture (e.g. DISCOVER codec), unless it is known for sure that the sequence to be encoded is characterised by low motion activity.

4.8 Conclusion

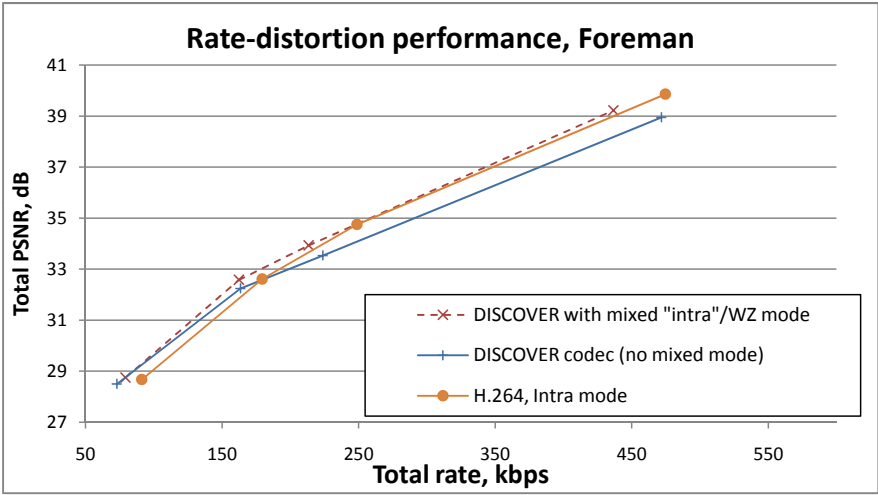
In this chapter we have addressed different problems concerning correlation modeling and model parameters estimation. The correlation model of the decoder has been studied thoroughly, its drawbacks and weak points have been identified, and an algorithm for improving the correlation model estimation has been proposed. Closed-form expressions for the minimal mean-squared error reconstruction at the decoder have also been derived in the first part of the chapter addressing decoder problems.

In the second part we presented a hybrid encoder/decoder rate control solution, which main goal is reducing the decoding complexity by diminishing the number of decoder requests. The encoder part of this hybrid solution estimates the minimal rate, while the sufficient rate is determined by the decoder, which employs the developed bit error rate estimation technique.

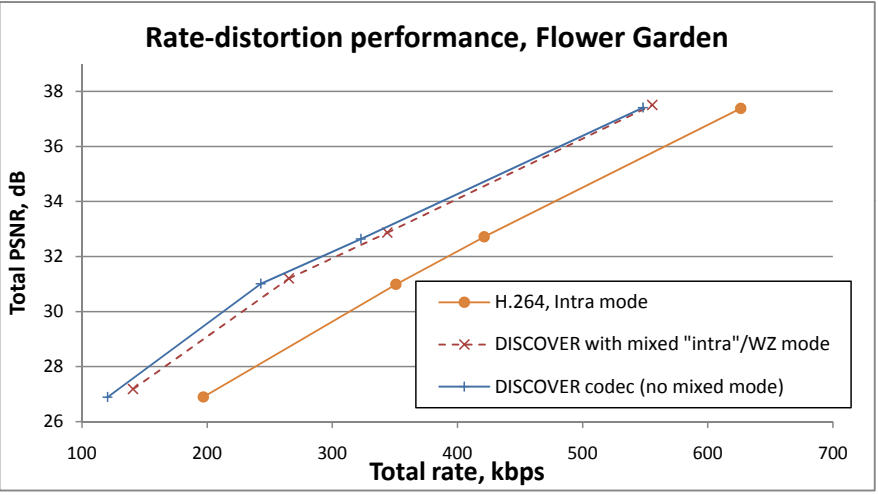
The last part concerns correlation modeling problems at the encoder side. In fact, determining the correlation model at the encoder requires violating the distributed coding principle stating that all frames are encoded independently, and no inter-frame processing is done on the encoder side. However, this principle limits the performance of real video coding systems, because video data usually cannot be modeled using models which are static in temporal and/or spatial domains.



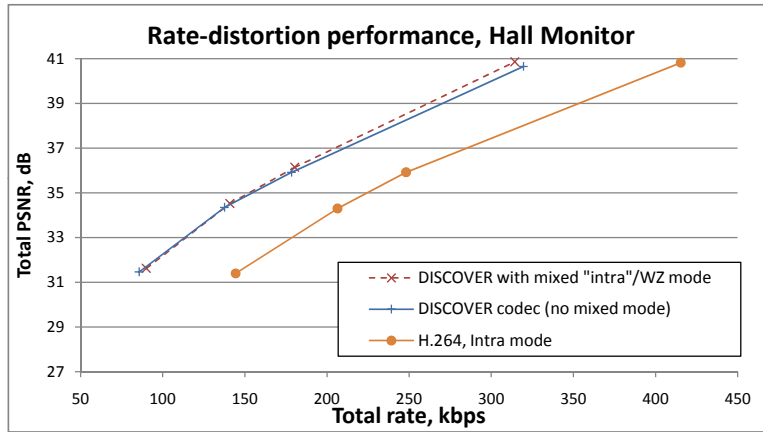
(a) Coastguard sequence, QCIF, 15Hz



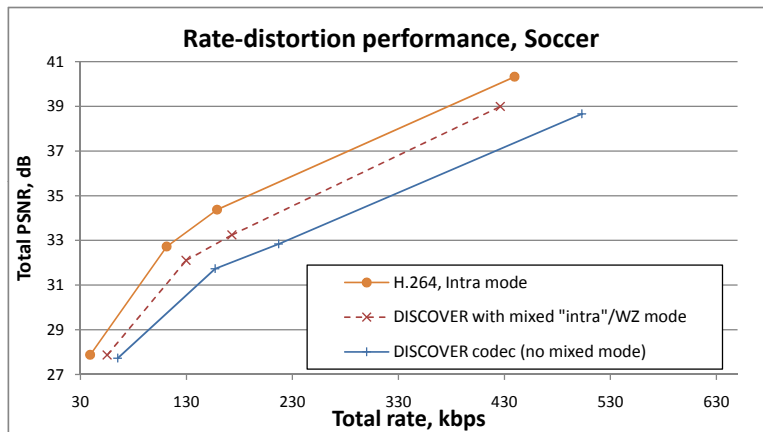
(b) Foreman sequence, QCIF, 15Hz



(c) FlowerGarden sequence, QCIF, 15Hz

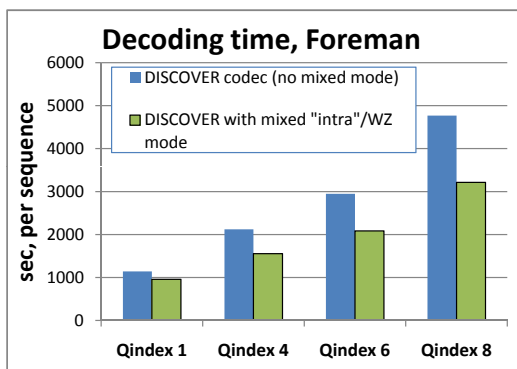


(d) Hall monitor sequence, QCIF, 15Hz

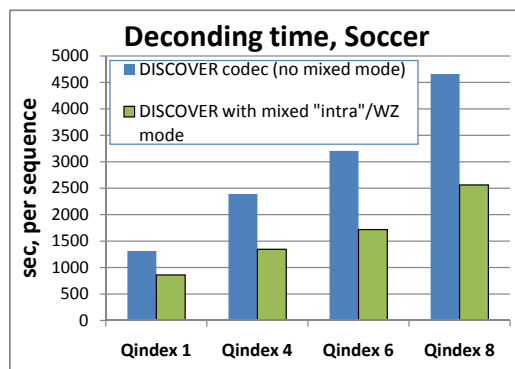


(e) Soccer sequence, QCIF, 15Hz

Figure 4.22: Rate-distortion performance with mixed "intra"/WZ encoding mode



(a) Foreman sequence, QCIF, 15Hz



(b) Soccer sequence, QCIF, 15Hz

Figure 4.23: Decoding time with mixed "intra"/WZ encoding mode

That's why a minimal inter-frame processing should be allowed at the encoder to achieve good rate-distortion performances. We have presented several algorithms to tackle the problem of correlation modeling, and using the model information by the encoder to improve the performance of the DISCOVER codec. Namely, an algorithm has been proposed to adaptively determine the quantisation table for Wyner-Ziv frames, and another one to perform a block-level decision about the encoding mode ("intra" or Wyner-Ziv) based on the estimation of the motion activity. Both proposed algorithms demonstrate rate-distortion performance improvements, but increase the encoder complexity.

However, there is still a lot of research to be done in the area of correlation modeling and model parameters estimation. For example, it is very important to reduce the gap between the rate-distortion performance of the "ideal" case when the model is perfectly known, and that one of a practical system (see Figure 4.3). A possible way is to improve the EM-based model refining algorithm described in Section 4.2, which is now mainly oriented on refining the global correlation model, and not the individual models per coefficient.

Another important issue is related to exploiting at the encoder the information about the correlation level in the most efficient way. Several different ways of using this information have been proposed, but an integrated solution is still to be developed. For example, the rate-distortion optimisation can be coupled with the frame-level ([ABP06]) and block-level (Section 4.7) "intra"/WZ mode decision techniques.

All these encoder side rate control solutions are very sensitive to the precision and actuality of the correlation model parameters estimation, because a failure on this stage cannot always be compensated by the decoder complement of the rate control. That is why developing algorithms like the one presented in Section 4.6 is very important.

Chapter 5

Source coding aspects of Wyner-Ziv video coding

Until now, we have been considering separate video frames as memoryless sources. In practice, however, there is a correlation, or memory, inside each frame. This correlation can be taken into account to improve the rate-distortion performance of a Wyner-Ziv coding system.

We address this problem on three different levels. First, we consider a replacement for the DCT transform, which allows us to decorrelate frequency bands more efficiently, thus reducing the residual memory in each bands. Namely we propose to use the oriented wavelet transform developed by Chappelier *et al.* [CG06]. This transform outperforms DCT, when the original frame contains a lot of structured texture information oriented at an arbitrary angle.

The second level is the level of DCT coefficients. Here, we develop a distributed DPCM (Differential Pulse Code Modulation) approach. The classical DPCM allows to reduce the energy of a sequence with memory by designing an appropriate prediction filter. In distributed DPCM, the filter design is done under an additional constraint on the correlation level between the source sequence and the side information sequence (both are predicted independently with possibly different filters).

For exploiting the statistical redundancy we propose to employ Huffman coding of quantisation indices instead of fixed-length coding used in DISCOVER. We also discuss the problem of the bitstream desynchronisation which raises due to the fact that bit planes do not have the same size anymore.

5.1 Exploiting memory on the pixel level

We start by treating the source memory problem at the highest level, i.e. the pixel level. A common way to reduce the memory on the pixel level, or spatial correlation in the pixel domain, in an image, is applying a transform, e.g. the discrete cosine

transform (DCT). The DCT transform and its analogues have been widely used in image and video compression (JPEG [PM92], H.264/AVC [WSBL03]) due to its nice decorrelation and energy compaction properties [RY90].

However, in recent years the discrete wavelet transform (DWT) has started to gain popularity in the research community, and even replaced the DCT transform in latest image coding standards (e.g. JPEG2000 [SCE01]). The advantage of the DWT for image coding is the absence of blocking artifacts in the decoded image, but also a better energy compaction due to the multi-resolution nature of the wavelet analysis, which allows for a more efficient coding [XROZ99].

That is why in this section we propose using a DWT instead of the currently adopted DCT-like integer transform in the DISCOVER codec, for better exploiting the spatial correlation inside Wyner-Ziv (WZ) frames. For this purpose, we have chosen the oriented wavelet transform (OWT) proposed by Chappelier *et al.* [CG06]. This transform is a lifted wavelet transform, in which the lifting steps of a one-dimensional wavelet basis are oriented along a discrete set of directions. The orientation in each position is chosen to minimise the energy of the wavelet coefficient in the high-frequency subband. Therefore, the geometry of the image is explicitly described by an orientation map, which is transmitted along with wavelet coefficients. This transform has been chosen because of its good energy compaction properties along the image contours, which are the most difficult areas for the Wyner-Ziv coding.

The section is organised as follows. First, we will outline the lifting wavelet decomposition scheme. Then, we will describe the oriented wavelet transform itself. The integration with the DISCOVER codec is described after, with some remarks concerning the choice of quantisation tables. Finally, rate-distortion performance results will be presented.

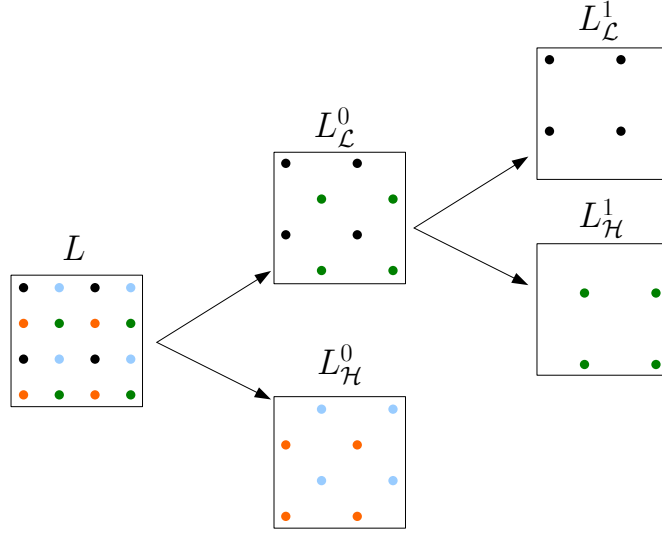
5.1.1 Lifting wavelet decomposition

The lifting scheme provides an efficient way of computing wavelet transforms [DS98]. The idea behind lifting is that the polyphase matrix of a wavelet filter bank allowing perfect reconstruction can be factorised in a finite number of predict and update steps, which are performed in-place (no need for memory allocation), followed by a final scaling step. The perfect reconstruction is achieved by simple inversion of different lifting steps.

For symmetric filters (e.g. the 9/7 and 5/3 wavelet filters commonly used in image compression), the lifting operations are also symmetric, and the wavelet analysis is performed by applying successively the following predict $P(\alpha_i)$ and update $U(\beta_i)$ operations:

$$P(\alpha_i) : S[n_1] \leftarrow S[n_1] + \alpha_i (S[n_1 - 1] + S[n_1 + 1]), \quad n_1 = 2n + 1, \quad (5.1)$$

$$U(\beta_i) : S[n_2] \leftarrow S[n_2] + \beta_i (S[n_2 - 1] + S[n_2 + 1]), \quad n_2 = 2n, \quad (5.2)$$

Figure 5.1: Quincunx decomposition of the grid L (two levels shown)

where α_i and β_i are the lifting factors of the predict and update step i . The final scaling step is then applied to obtain properly normalised coefficients:

$$R(\chi) : \begin{cases} S[n_1] \leftarrow \frac{1}{\chi} S[n_1], & n_1 = 2n + 1, \\ S[n_2] \leftarrow \chi S[n_2], & n_2 = 2n. \end{cases} \quad (5.3)$$

An approximation of the signal S is obtained at even positions $n_2 = 2n$ of the filtered signal, and corresponds to a low-pass filtering operation followed by down-sampling. Odd positions $n_1 = 2n + 1$ contain the prediction error, corresponding to a high-pass filtering operation followed by down-sampling.

The original signal S can be reconstructed by inverting the lifting steps, i.e. by applying successively:

$$R\left(\frac{1}{\chi}\right), U(-\alpha_{N-1}), P(-\beta_{N-1}), \dots, U(-\alpha_0), P(-\beta_0). \quad (5.4)$$

5.1.2 Oriented wavelet transform

Let us first define the *quincunx* decomposition which will be used in the oriented wavelet transform (OWT).

Two successive levels of a quincunx decomposition of a square sampling 2D grid L (e.g. image) are depicted on Figure 5.1. On the first level the positions (i, j) of L with even $(i + j)$ form a sub-grid L_L^0 , while the other positions (for which $(i + j)$ is odd) form a complementary sub-grid L_H^0 , so that:

$$L = L_L^0 \cup L_H^0. \quad (5.5)$$

Note that both $L_{\mathcal{L}}^0$ and $L_{\mathcal{H}}^0$ are *quincunx* patterns (as opposed to the *square* pattern of the original L).

On the second level the same operation is applied to the sub-grid $L_{\mathcal{L}}^0$ rotated by $\pi/4$, which amounts to selecting in $L_{\mathcal{L}}^1$ those positions for which both i and j (coordinates in the space of the grid L) are even, and putting the other positions into the complementary sub-grid $L_{\mathcal{H}}^1$, so that:

$$L_{\mathcal{L}}^0 = L_{\mathcal{L}}^1 \cup L_{\mathcal{H}}^1. \quad (5.6)$$

Here $L_{\mathcal{L}}^1$ and $L_{\mathcal{H}}^1$ are again *square* patterns. Therefore, the first level decomposition can be again applied to $L_{\mathcal{L}}^1$ to obtain $L_{\mathcal{L}}^2$ and $L_{\mathcal{H}}^2$, followed by the second level decomposition etc., till the step number $l - 1$, forming an l -level *quincunx pyramid*. This pyramid gives a multiresolution representation of a 2D signal, which is used to define the oriented wavelet transform.

The wavelet coefficients, corresponding to the prediction error, are computed on $L_{\mathcal{H}}^k$ grids, while the approximation of the image (corresponding to the low-pass filtered signal) is computed on $L_{\mathcal{L}}^k$ grids, on which the decomposition is reiterated. Each sample $n \in L_{\mathcal{H}}^k$ has four neighbours in $L_{\mathcal{L}}^k$, and therefore can be predicted from any combination of these samples. In contrast to lifted quincunx wavelets where the prediction is done using all four samples, Chappelier *et al.* propose to choose one of the two possible directions (horizontal and vertical for a *quincunx* grid $L_{\mathcal{L}}^k$, or diagonal and antidiagonal for a *square* grid $L_{\mathcal{L}}^k$), and predict the sample in n using only the two neighbours in the chosen direction, i.e. using 1D wavelet transform. This direction, or orientation, can be chosen to minimise the prediction error, which is the best choice in terms of energy compaction. However, if the transmission cost of the orientation map $m_{L_{\mathcal{H}}^k}$ must also be taken into account, this choice might be sub-optimal. Therefore, a regularisation procedure for the quad tree map representation is proposed in [CG06], which optimises the map transmission rate and the energy compaction properties of the transform.

The wavelet coefficients at positions $n_1 \in L_{\mathcal{H}}^k$ are then computed using the following predict step k of 1D lifting:

$$P(\alpha_k) : \begin{cases} S[n_1] \leftarrow S[n_1] + \alpha_k \sum_{n_h \in R(n_1)} S[n_h], & \text{if } m_{L_{\mathcal{H}}^k} = 0, \\ S[n_1] \leftarrow S[n_1] + \alpha_k \sum_{n_v \in C(n_1)} S[n_v], & \text{if } m_{L_{\mathcal{H}}^k} = 1, \end{cases} \quad (5.7)$$

where $R(n_1)$ and $C(n_1)$ represent respectively the set of the two horizontal and two vertical neighbours of n_1 in $L_{\mathcal{L}}^k$.

The update lifting steps also need to be modified. Indeed, each sample in $L_{\mathcal{L}}^k$ can be used from 0 to 4 times for predicting other samples in $L_{\mathcal{H}}^k$, and the orthogonality property is lost (in classical 1-D wavelet transform each sample is used exactly 2 times). Chappelier *et al.* [CG06] propose the following rule of modifying the update

step coefficients β_k to preserve statistical properties of the low-frequency band:

$$\beta_k^*[n_2] = \begin{cases} \frac{v}{2}\beta_k, & \text{if } v \neq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (5.8)$$

where v is the number of times the sample at position $n_2 \in L_{\mathcal{L}}^k$ has been used for prediction.

The update step is then defined as follows:

$$U(\beta_k) : S[n_2] \leftarrow S[n_2] + \beta_k^*[n_2] \sum_{n \in \mathcal{U}(n_2)} S[n], \quad (5.9)$$

where $\mathcal{U}(n_2)$ is the set (possibly empty) of all positions in $L_{\mathcal{H}}^k$ using the sample at position $n_2 \in L_{\mathcal{L}}^k$ as a predictor. Please note that if the orientation map is homogeneous (i.e. the direction is the same for all points in $L_{\mathcal{H}}^k$), then $\beta_k^*[n_2] = \beta_k \forall n_2 \in L_{\mathcal{L}}^k$, and the decomposition is equivalent to the 1-D wavelet decomposition applied along the selected direction.

The perfect reconstruction is guaranteed by inverting each lifting step, with the knowledge of the orientation map for all decomposition levels from 0 to $l-1$:

$$m = \bigcup_{k=0}^{l-1} m_{L_{\mathcal{H}}^k}. \quad (5.10)$$

This orientation map thus needs to be transmitted to the decoder.

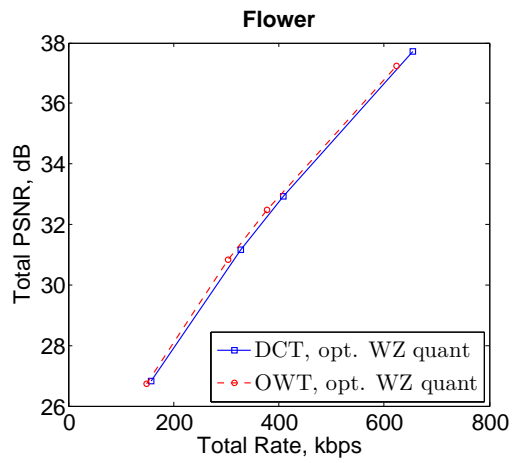
5.1.3 Experimental results with the DISCOVER codec

To evaluate the coding rate-distortion performance with the OWT, we have replaced the DCT in the DISCOVER codec with the OWT. We use the quad tree optimisation approach proposed in [CG06] for the orientation map, which needs to be transmitted to the decoder. This optimisation aims at solving the two problems:

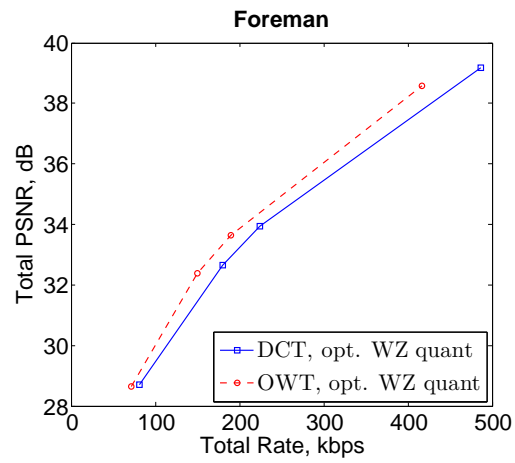
- reducing the rate cost of the map;
- regularising the map so that it becomes piece-wise homogeneous, making the transform orthogonal (or nearly orthogonal), which is important for the rate-distortion optimisation in transform domain (see below).

After the optimisation, the map is encoded with a CABAC encoder [MSW03].

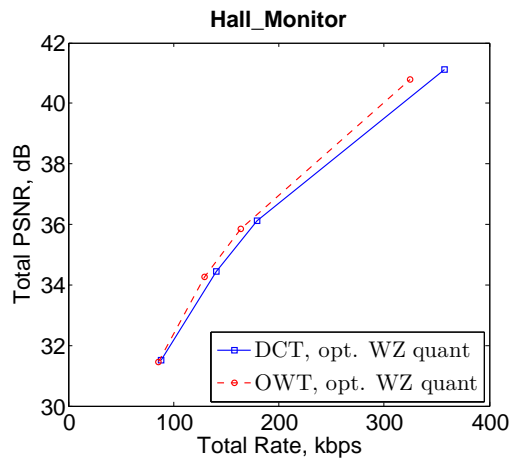
The implementation of the DISCOVER codec has had to undergo some modifications because of the fact that the size of different transform bands is different (with DCT all the bands have the same size). However, no other important changes have been required to the algorithm to adopt the new transform. In particular, the correlation model has been left the same as used with the DCT transform, i.e. zero-mean



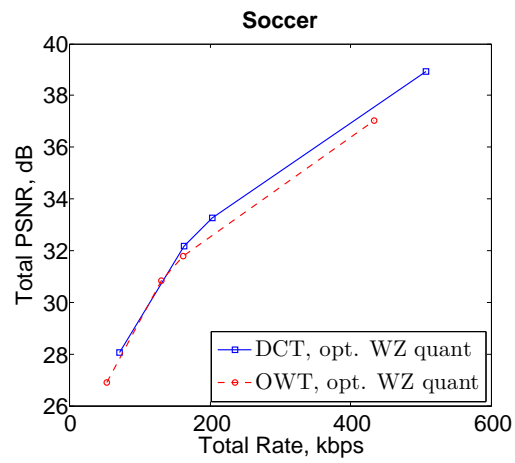
(a) FlowerGarden sequence, QCIF, 15Hz



(b) Foreman sequence, QCIF, 15Hz



(c) Hall Monitor sequence, QCIF, 15Hz



(d) Soccer sequence, QCIF, 15Hz

Figure 5.2: Rate-distortion performance with oriented wavelet transform (OWT) and DCT (optimised quantisation table for WZ frames is used)

Laplacian with parameter α (one per band or per coefficient), see Section 4.1 for more details.

However, the parameters of the WZ frames encoding must have been reviewed, namely the WZ quantisation table. To ensure an optimal in the rate-distortion sense selection of the number of quantisation levels per band, in our experiments below we use the method proposed in Section 4.5 for both DCT- and OWT-based coding schemes. Please note that for low-complexity coding fixed quantisation schemes can be defined similarly to the DCT case (as described in Section 2.3.4.2), but we have not done this work yet.

Figure 5.2 shows rate-distortion performance results for different QCIF sequences at 15Hz. As usual, the total rate includes the rate for key frames and WZ frames, and also includes the cost of the orientation map transmission in the case of the OWT, while the total PSNR is the average PSNR for all decoded frames including key frames and WZ frames.

Looking at Figure 5.2 one can observe that the performance differs for different sequences. We relate this behaviour to the amount of structured texture information contained in the video sequence. Indeed, the Foreman and the Hall Monitor sequences contain more diagonal edges and object contours than the other two sequences, and the OWT has better energy compaction properties along such diagonal edges. That is why using the OWT for those sequences is a significant advantage to the coding system (resulting in up to ≈ 0.8 dB gain). On the other hand, in the FlowerGarden sequence a large portion of the frame is covered with an unstructured yet having high contrast texture (flower bed), which makes the orientation map very noisy, and thus expensive to code, penalising the total coding performance. For the Soccer sequence no high gains are expected, as it does not contain much structured details. Instead, it contains a lot of smooth areas with low luminance variation, for which the DCT is very efficient.

5.2 Distributed DPCM

In this section we tackle the problem of the memory inside a transform band. We concentrate on the prediction in the DC band, first because the dependencies between the low-frequency coefficients are stronger than those between the high-frequency ones, and second because the coding of the DC band usually requires the most rate, and thus the gain from using an appropriate prediction, should be substantial. This work has been done jointly with Jayanth Nayak, and has been published in [NRKG08].

We will start by recalling the principles of the classical Differential Pulse Code Modulation (DPCM) approach in a single source case. After that we will extend it to the Wyner-Ziv coding, i.e. when both the source and the side information need to be predicted. We call this approach Distributed DPCM (DDPCM), or Wyner-Ziv DPCM (WZDPCM). Finally, we present experimental results for Gauss-Markov sources as

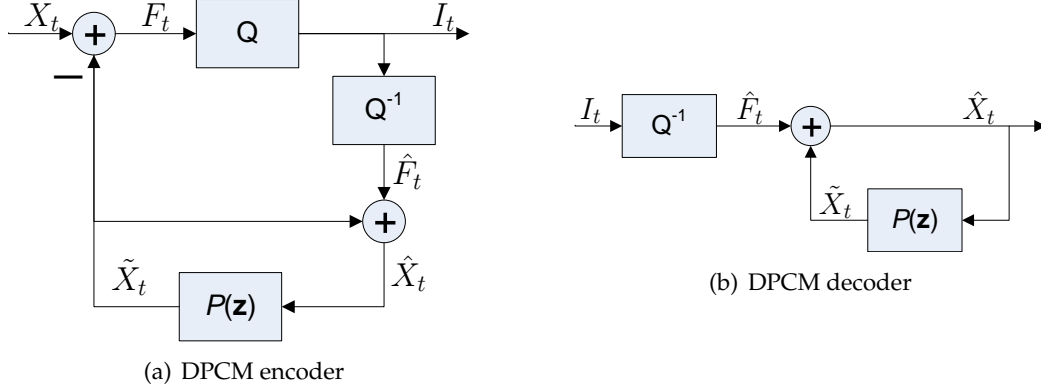


Figure 5.3: DPCM coding scheme

well as for real video data, which confirm the efficiency of the proposed approach.

5.2.1 DPCM

Let us consider a real-valued random variable X with zero mean, and let us denote with X_t a realisation of this variable at time instant t , so that the sequence X_t , $t \geq 0$, defines a discrete-time random process. The autocorrelation function between two time instants t and $t - i$ (supposing that X_t is a second-order stationary) of the process X_t is given by:

$$R_{XX}(i) = \mathbb{E}[X_t X_{t-i}]. \quad (5.11)$$

Given that we are allowed a rate of R bits/sample to compress the process, we wish to minimise the mean square error (MSE) between the source and its reconstruction based on the compressed bits. DPCM is a technique that was developed for compressing such sources [Nol78] and has found application in a number of practical scenarios.

A DPCM coding system is depicted on Figure 5.3. It is defined by a prediction filter $P(\mathbf{z})$ and a quantiser Q . The filter $P(\mathbf{z})$ is used to predict the current value X_t based on reconstructions of past samples $\hat{X}_{t-1}, \dots, \hat{X}_{t-p}$, where p is the prediction order. The prediction error $F_t = X_t - P(\hat{X}_{t-1}, \dots, \hat{X}_{t-p})$ is quantised and transmitted to the decoder. For simplicity, the quantiser is generally chosen to be uniform. This prediction structure, where the quantised residual is used in prediction, is called *closed loop* (as opposed to the *open loop*, where the unquantised residual is used in prediction).

While computing the optimal in terms of the distortion-rate (DR) performance prediction filters is an unsolved problem, at high rates, the DR performance of DPCM can be approximated by the DR function of a memoryless version of the residual. The latter can be further bounded by the DR function $D_G(R)$ of a memoryless Gaussian source with the same variance σ_F^2 as the residual F :

$$D_G(R) = \frac{\sigma_F^2}{2^{2R}}. \quad (5.12)$$

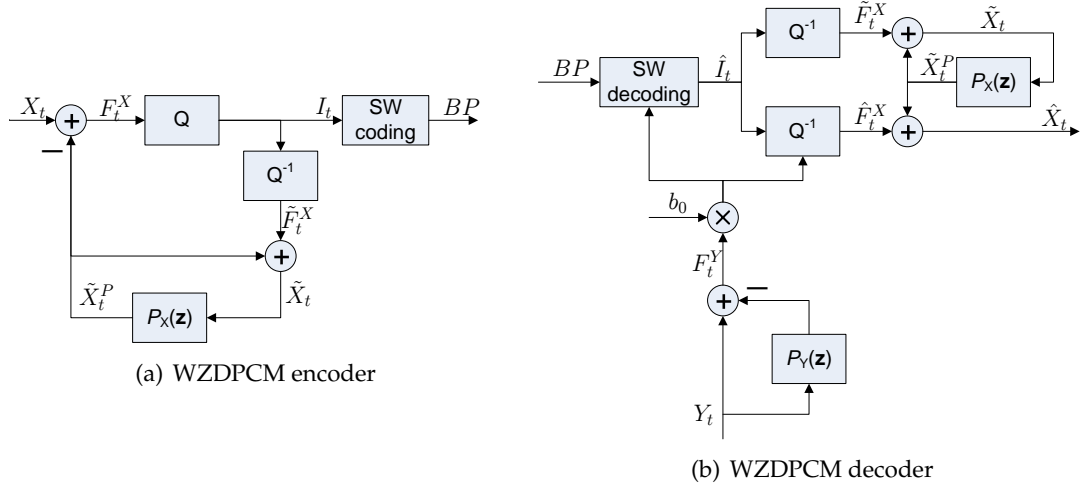


Figure 5.4: WZDPCM coding scheme

Therefore, a filter that minimises the variance σ_F^2 of the residual, is a good choice. Coefficients of the optimal in terms of the variance σ_F^2 prediction filter $P(\mathbf{z}) = \sum_{i=1}^p a_i z_i$ of order p are given by:

$$\begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} R_{XX}(0) & \dots & R_{XX}(p-1) \\ \vdots & \ddots & \vdots \\ R_{XX}(-p+1) & \dots & R_{XX}(0) \end{bmatrix}^{-1} \begin{bmatrix} R_{XX}(-1) \\ \vdots \\ R_{XX}(-p) \end{bmatrix}. \quad (5.13)$$

5.2.2 Prediction in Wyner-Ziv coding

Now let us consider a Wyner-Ziv coding scheme. Let (X_t, Y_t) be the pair of source-side information processes, which are correlated and have memory. Our scheme of a Wyner-Ziv coding system using DPCM is depicted on Figure 5.4. In this scheme, X_t and Y_t are both predicted, with possibly distinct filters $P_X(\mathbf{z})$ and $P_Y(\mathbf{z})$ (even the filters orders p_X and p_Y need not to be the same). The source X_t is predicted in closed loop, while the side information Y_t need not be quantised, thus predicted in open loop. To get the best linear estimate of the source residual F_t^X , the side information residual F_t^Y is scaled with a factor b_0 :

$$b_0 = \rho_F \frac{\sigma_{F^X}}{\sigma_{F^Y}}, \quad (5.14)$$

where σ_{F^X} (σ_{F^Y}) is the variance of the residual F^X (F^Y respectively), and ρ_F is the correlation coefficient between F_t^X and F_t^Y :

$$\rho_F = \frac{\mathbb{E}[F_t^X F_t^Y]}{\sigma_{F^X} \sigma_{F^Y}}. \quad (5.15)$$

The source residual F_t^X is then uniformly quantised, and the quantisation index I_t is coded with a Slepian-Wolf encoder (e.g. a turbo encoder).

The decoder first decodes (estimates) the quantisation index \hat{I}_t , using $b_0 F_t^Y$ as the side information. After that, two distinct estimations of the quantised residual F_t^X are generated:

1. \tilde{F}_t^X is an estimation based *only* on the estimation \hat{I}_t of the quantisation index; it is used for the prediction filter state update in a way the encoder does, therefore without using the side information:

$$\tilde{X}_t = P_X(\tilde{X}_{t-1}, \dots, \tilde{X}_{t-p_X}) + \tilde{F}_t^X, \quad (5.16)$$

where \tilde{X}_t is the estimation of X_t for the prediction filter;

2. \hat{F}_t^X is an estimation based on \hat{I}_t and the side information $b_0 F_t^Y$; this value is used for constructing the final estimate \hat{X}_t of the symbol X_t :

$$\hat{X}_t = P_X(\tilde{X}_{t-1}, \dots, \tilde{X}_{t-p_X}) + \hat{F}_t^X. \quad (5.17)$$

Design of prediction filters for X_t and Y_t requires first defining a cost function J to minimise. Following the idea of the classical DPCM, a separate prediction can be done, meaning $J = \sigma_{F^X} + \sigma_{F^Y}$. However, and this is confirmed by our simulations, this choice is not the best for Wyner-Ziv coding, as separately designed filters can significantly reduce the correlation between F_t^X and F_t^Y .

Assuming high rate quantisation, optimal Slepian-Wolf coding, and approximating the rate-distortion function by that one of a pair of *memoryless* sources, Nayak derives [NRG08] the following cost function for a pair of Wyner-Ziv prediction filters:

$$J_{WZ} = \sigma_{F^X}^2 (1 - \rho_F^2). \quad (5.18)$$

Indeed, this function is intuitively satisfying, as we can improve the rate-distortion performance either by reducing the variance of the residual F_t^X , thereby improving the quantisation, or by increasing the correlation ρ_F of the residuals F_t^X and F_t^Y , thereby improving the Slepian-Wolf coding.

The coefficients a_i^X and a_j^Y of J_{WZ} -optimal filters can be obtained using the equivalence of this cost function J_{WZ} with that one of the least squares cost function [NRG08] for the problem of the best linear estimation of X_t given a vector of predictors $Z_t^T = [X_{t-1}, \dots, X_{t-p_X}, Y_t, \dots, Y_{t-p_Y}]$, i.e.:

$$\begin{bmatrix} c_1 & c_2 & \dots & c_{p_X+p_Y+1} \end{bmatrix}^T = \left[\mathbb{E} [Z_t Z_t^T] \right]^{-1} \mathbb{E} [X_t Z_t], \quad (5.19)$$

$$a_i^X = c_i, \quad i = 1, \dots, p_X, \quad (5.20)$$

$$a_j^Y = \frac{c_{j+p_X+1}}{c_{p_X+1}}, \quad j = 1, \dots, p_Y, \quad c_{p_X+1} \neq 0, \quad (5.21)$$

$$b_0 = c_{p_X+1}. \quad (5.22)$$

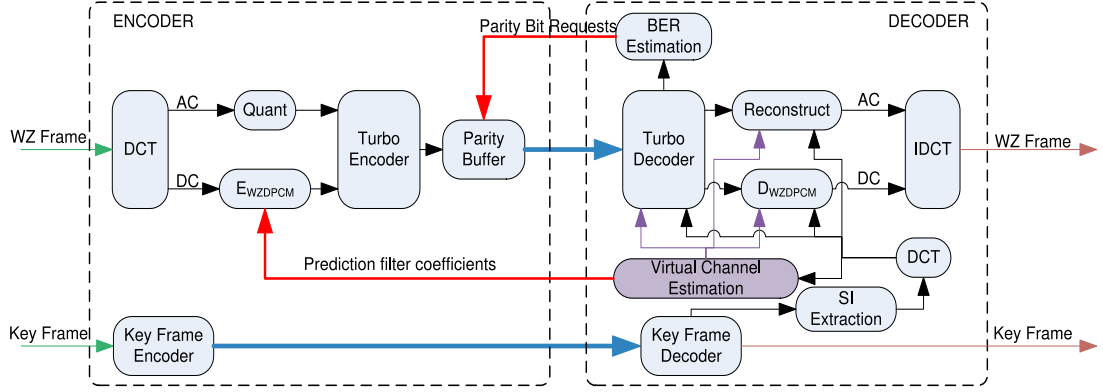


Figure 5.5: DISCOVER codec with distributed DPCM on the DC band

Please note that the scaling coefficient b_0 is also obtained as a byproduct of (5.19). If it is zero, delay elements would be needed at the input of the side information loop [NRG08].

5.2.3 Application to video

For evaluation purposes, the proposed distributed prediction method has been integrated into the DISCOVER coding software. In our experiments, the prediction was used only for the DC band of Wyner-Ziv (WZ) frames. The sequence X_t (Y_t) is obtained by raster scanning the DC band of the WZ frame (side information frame respectively). The modified DISCOVER architecture is depicted on Figure 5.5.

Both prediction filters $P_X(\mathbf{z})$ and $P_Y(\mathbf{z})$ are designed by the decoder that starts by creating the side information and extracting its DC band Y_t (prior to transmission of any information about the Wyner-Ziv frame). Then the two prediction filters are created using (5.19), and the coefficients a_i^X , $i = 1, \dots, p_X$, are communicated to the encoder via the return channel. Please note that the decoder does not have access to X_t required to compute (5.19). It is therefore approximated with $X_t \approx Y_t + R_t$, where R_t is the DC band of the motion compensated residual. The same approach is used by Brites *et al.* [BP08] who use this approximation $R_t \approx X_t - Y_t$ for estimating correlation model parameters (see also Section 2.3.4.6).

Having received the prediction filter coefficients, the encoder starts encoding of DC samples X_t as described above, i.e. it computes the closed loop residual F_t^X and quantises it uniformly. Quantisation indices I_t are then encoded using the standard procedure in DISCOVER (see Section 2.3.4.3), bit plane by bit plane (the most significant bit plane is transmitted first). To reduce the decoding delay, AC bands could be processed (encoded and transmitted) before the DC band, as the encoding of the latter has to wait until the prediction filter coefficients are received from the decoder via the return channel.

5.2.4 Experimental results

We will first present experimental results for the proposed distributed prediction algorithm being applied to synthetic data (Gauss-Markov sources), and after that show rate-distortion performances for the real video data.

5.2.4.1 Experimental results with a Gauss-Markov process

A second order Gauss-Markov process used in the simulation, is a process that satisfies:

$$\begin{bmatrix} X_t \\ Y_t \end{bmatrix} = - \sum_{i=1}^2 \mathbf{B}_i \begin{bmatrix} X_{t-i} \\ Y_{t-i} \end{bmatrix} + \begin{bmatrix} N_t^X \\ N_t^Y \end{bmatrix}, \quad (5.23)$$

where \mathbf{B}_i , $i = 1, 2$, are the 2×2 matrices defining the autoregressive process, and $\{(N_t^X, N_t^Y)\}_{t=-\infty}^{\infty}$ is a memoryless stationary Gaussian noise process. For our experiment below we have chosen the matrices \mathbf{B}_i , $i = 1, 2$ such that the correlation between X_t and Y_t had approximately the same level as the correlation between the mean-removed DC band of the Foreman sequence with the respective DC band of the side information. We compare the following methods:

1. **No prediction** - standard DISCOVER codec as described in Section 2.3.4;
2. **WZDPCM with traditional separate (TS) prediction** filters: $P_X(\mathbf{z})$ and $P_Y(\mathbf{z})$ are designed independently, each minimising the respective variance of the residual. In other words, the cost function which is minimised is $J_{\text{TS}} = \sigma_{F^X} + \sigma_{F^Y}$. The correlation between F_t^X and F_t^Y is therefore ignored in the filter design.
3. **WZDPCM with Wyner-Ziv (WZ) prediction** filters: $P_X(\mathbf{z})$ and $P_Y(\mathbf{z})$ are designed to minimise the Wyner-Ziv cost function (5.18); their coefficients are given by (5.19);
4. **Wyner-Ziv bound**: we also plot for information the Wyner-Ziv bound for memoryless jointly Gaussian sources with the same second-order statistics as the prediction residual in both TS and WZ prediction. The WZ bound is given by (1.48).

The various cost functions obtained using the two prediction filter design techniques, are given in Table 5.1. Note that the separate filter design (TSP) reduces the variance $\sigma_{F^X}^2$ significantly, but also reduces the correlation between F_t^X and F_t^Y . With the Wyner-Ziv prediction filter design (WZP) the correlation is also slightly reduced, but this decrease is compensated by the decrease of the source variance $\sigma_{F^X}^2$. The corresponding rate-distortion performance for the Gauss-Markov process is shown on Figure 5.6. These results totally agree with the cost function analysis. Namely, the Wyner-Ziv prediction gains around 1.06 dB over no prediction (at high bit rates), while TSP does not show any gain at all, exactly as predicted by the WZ cost function J_{WZ} in Table 5.1.

Table 5.1: Gauss-Markov process: Theoretical performance.

Prediction Type	Order	J_{TS}	J_{WZ}	σ_{EX}^2	ρ_F
No prediction	—	48.99	30.89	46.05	0.985
Traditional separate prediction (TSP)	1	41.61	30.91	38.95	0.918
	2	41.52	30.94	38.88	0.916
Wyner-Ziv prediction (WZP)	1	44.89	29.91	42.02	0.969
	2	46.15	29.52	43.24	0.979

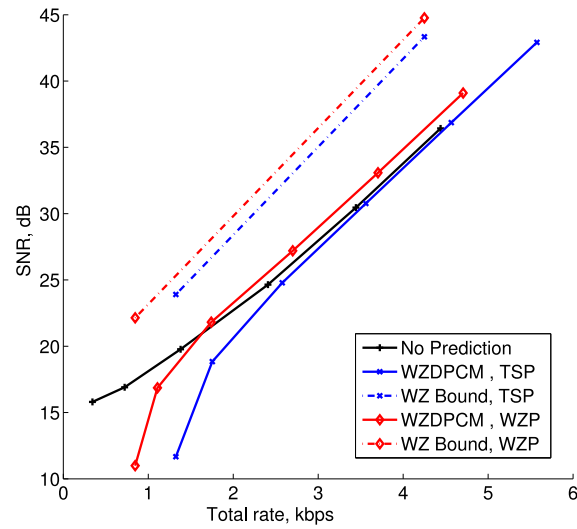


Figure 5.6: Rate-distortion performance with WZDPCM for a Gauss-Markov process

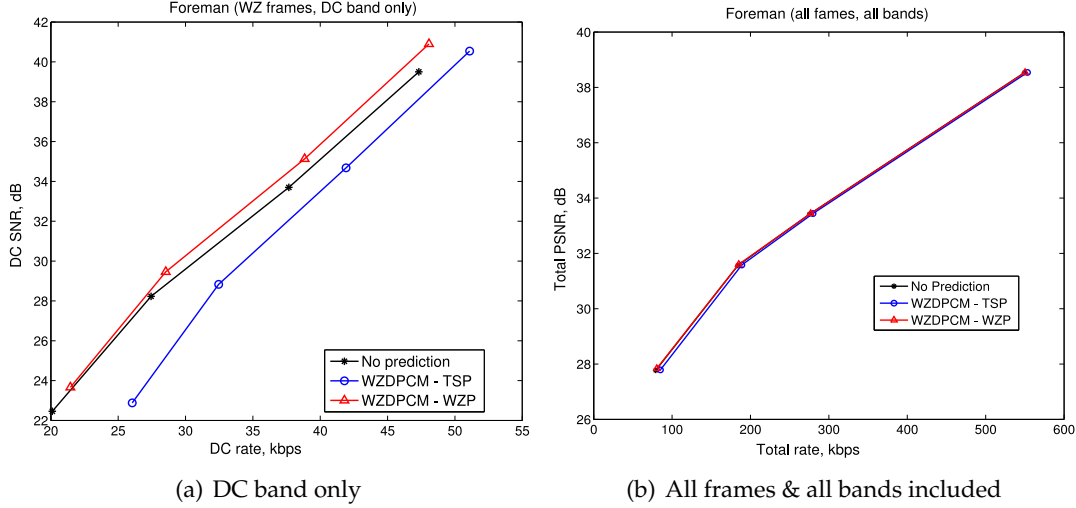


Figure 5.7: Rate-distortion performance with WZDPCM for Foreman sequence

5.2.4.2 Experimental results with real video data

Rate-distortion results for the Foreman sequence are shown on Figure 5.7. The figure on the left shows the performance for the DC band only. Here the gain of the proposed technique (WZDPCM - WZP) over no prediction is about 1 dB at high rates. However, this gain is barely visible when the performance of all bands is taken into account, and averaged with the performance on key frames (see Figure 5.7(b)). The situation can be possibly improved by using the prediction for AC bands also, and not only for the DC band. Another potential improvement consists in making the 2D prediction instead of 1D, i.e. using in prediction previous samples which are adjacent to the current sample in horizontal *and* vertical directions (regarding the band as a 2D matrix of samples).

5.3 Huffman coding of quantisation indices

In the section we treat the problem of statistical redundancy at the level of quantisation indices. Because of the energy compaction property of the transform (e.g. DCT, or OWT), many coefficients of high frequency bands are concentrated around zero. Therefore, a source code (e.g. Huffman code) can be used for compressing quantisation indices more efficiently than a fixed-length code (FLC) adopted in DISCOVER.

In a Wyner-Ziv coding scheme, however, it is not obvious that a compression of quantisation indices can have any impact on the SW codec performance. Indeed, in theory the rate is given by the conditional entropy $H(Q|Y)$ of the quantisation index Q knowing the side information Y :

$$R_{\text{SW}} = H(Q|Y), \quad (5.24)$$

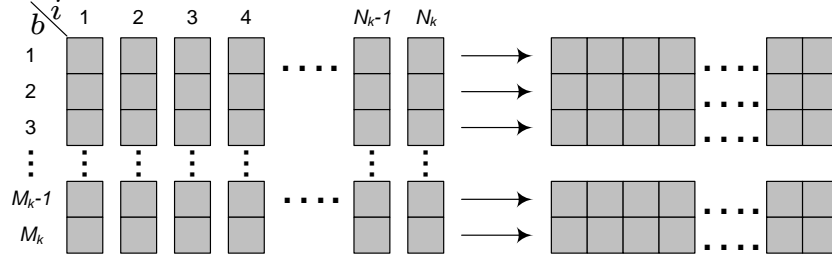


Figure 5.8: Bit plane extraction, FLC

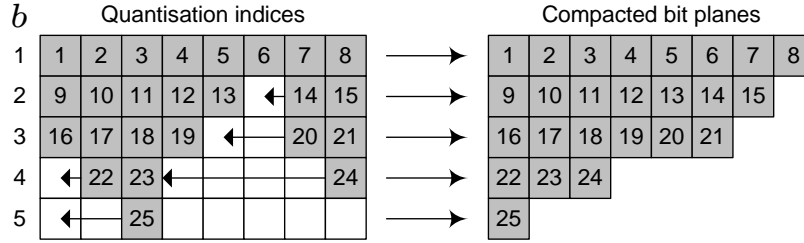


Figure 5.9: Bit plane compaction

thus the binarization code for Q doesn't matter. In practice, however, the following model would be more accurate [RMZG03]:

$$R_{\text{SW}} = aH(Q) + bH(Q|Y), \quad a, b > 0, \quad a + b = 1, \quad (5.25)$$

and therefore decreasing $H(Q)$ can lead to a decrease of R_{SW} (unless the respective increase of $\frac{b}{a}H(Q|Y)$ is greater). Our experimental results below confirm that the total rate of a practical Wyner-Ziv coding system (the DISCOVER codec) is decreased when quantisation indices are compressed with a Huffman code.

5.3.1 Integrating VLC support in DISCOVER coding scheme

The DISCOVER architecture uses a FLC for the quantisation indices binarization. That means that each quantisation index q_i^k of band k at position i is represented using exactly M_k bits. All bits with the same number b , $1 \leq b \leq M_k$, form the b -th bit plane of the band k , as shown on Figure 5.8, so that all bit planes have the same length N_k as the length of the band.

Let us now consider a VLC for the quantisation index binarization. In this case the number of bits in each bit plane b , $2 \leq b \leq M_k^{\text{vlc}}$, can be less than N_k (we denote with M_k^{vlc} the maximum VLC codeword length). To get continuous bit streams, they are compacted as shown for example on Figure 5.9, and encoded using the turbo encoder. Note that if the decoder has decoded all previous bit planes b_p , $1 \leq b_p < b$, without

errors, then it can determine the position of each bit of the bit plane b by making the inverse compaction.

The above approach with bit plane compaction has the following drawbacks:

- compacted bit planes can be too short for an efficient turbo coding, and
- a decoder desynchronisation is possible if previous bit planes were decoded with errors. This can lead to a wrong inverse compaction, and eventually to a wrong decoding of all subsequent bit planes.

To reduce the effect of turbo codes inefficiency for short sequences, we limit the minimal compacted bit plane length to be turbo encoded. If the bit plane is shorter than this minimal length, it is transmitted as it is, i.e. without encoding, and the CRC check-sum (see Section 4.4.2) for it is not transmitted. Another possible solution which has not yet been tested, is using block codes, or an arithmetic coder (e.g. CABAC), instead of turbo codes, for short bit planes.

Regarding the desynchronisation problem, we note that the hybrid rate control described in Section 4.4, is designed so that to ensure zero error rate at the turbo decoder issue (due to the usage of the CRC check). Therefore, the desynchronisation is only possible if this rate control fails. According to our experiments, this does not happen very often, but happens more often with short bit sequences than with long ones. Thus only less frequent symbols are usually concerned by the desynchronisation and incorrect reconstruction, which limits the distortion. We think however that the problem needs further investigation. For example, one possible improvement we see is applying a soft symbol-based decoding algorithm after all bit planes are decoded.

5.3.2 Experimental results

The proposed approach has been evaluated within the DISCOVER coding architecture, and rate-distortion results are presented on Figure 5.10. In this experiment, the distribution law of quantisation indices have been estimated using the reconstructed version of the closest key frame, which is available both at the encoder and the decoder. Thereby, the Huffman codetree, constructed using this estimated law, is exactly the same at both sides, and does not need to be transmitted. The most significant bit planes which have the length N_k are coded with a LDPC code, while remaining bit planes (which are shorter than N_k) are coded with turbo codes. The reason is that sequence-length dependent LDPC matrices are generated offline for a limited set of possible length (e.g. 1584, 6336 bits), and cannot be generated on-the-fly for an arbitrary length with our current software.

This experiment is not fully fair, as we used original data at the decoder to determine the correct compaction pattern. This has been done for the implementation simplicity, to avoid inserting re-synchronisation marks between different bands into

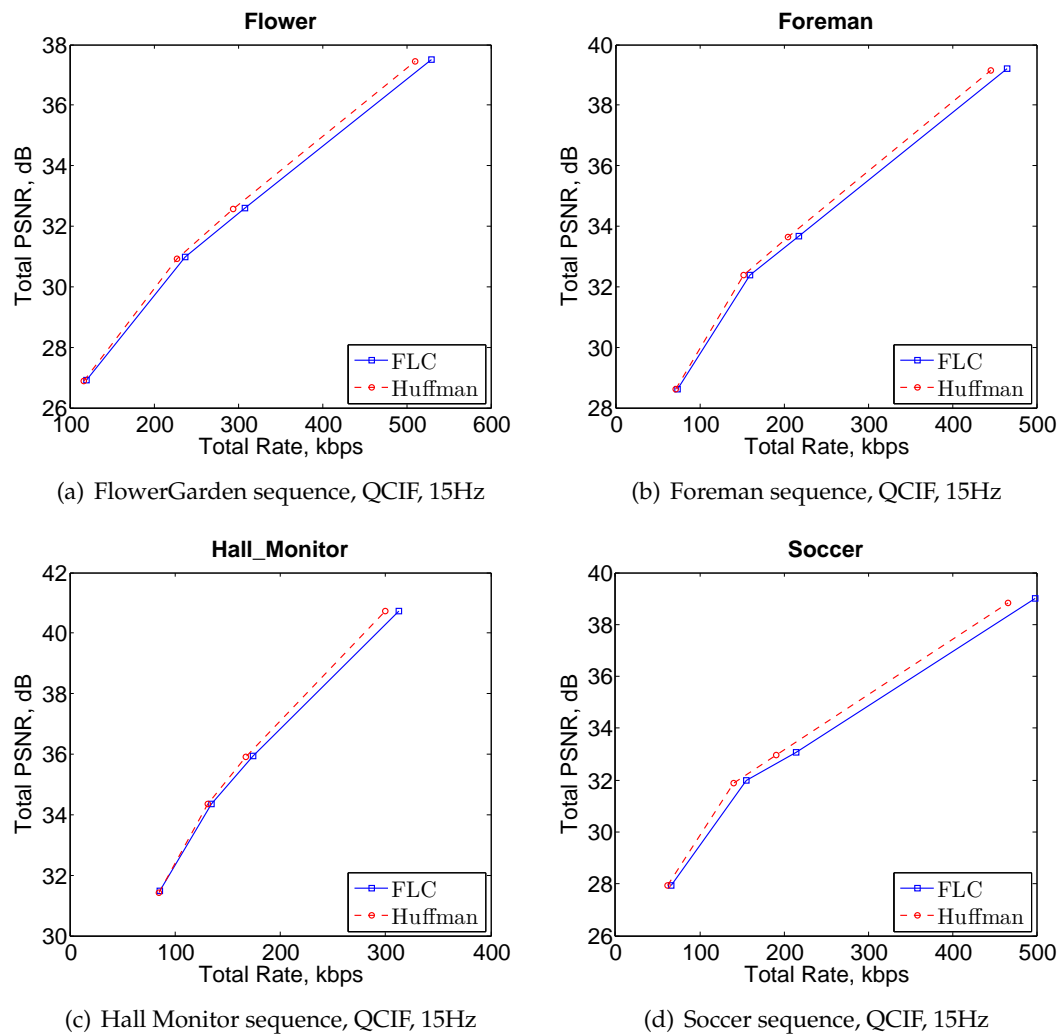


Figure 5.10: Rate-distortion performance with FLC and Huffman coding of quantisation indices

the bit buffer, which emulates the on-line encoding/decoding behaviour in the DISCOVER codec implementation. In a real on-line implementation, with the encoder working interactively with the decoder, this re-synchronisation can easily be achieved by issuing a special decoder request. Please note that only the compaction pattern construction is not fair: bit errors in erroneously decoded bit planes are not corrected, and respective symbols are incorrectly reconstructed in the decoded frame. That is why the reconstruction quality with the Huffman coding of quantisation indices is slightly lower than that one with the FLC. This decrease is however recouped by a significant gain in rate, as demonstrates Figure 5.10.

5.4 Conclusion

In this chapter we have addressed the problem of Wyner-Ziv coding of sources with memory, on four different levels.

First, we have considered using the oriented wavelet transform (OWT) [CG06] instead of the DCT which is used in DISCOVER. This transform has been shown to be very effective when the original frame contains a lot of structured texture information oriented at an arbitrary angle, and in general has better energy compaction properties than DCT. This is proved by our experiments, showing significant gains in rate-distortion performance for most video sequences, when the OWT is used.

Secondly, we addressed the memory problem on the level of DCT coefficients. We have developed a method for distributed prediction in Wyner-Ziv coding, which consist in designing the filters for the source and the SI with the special cost function taking into account both the variance of the prediction residual, and the correlation between the source and the side information. The gain of almost 1 dB is obtained when applying this method on the DC band of WZ frames, which is however marginal on total rate-distortion performance graphs, showing the performance for all bands and all frame types (Wyner-Ziv and key frames).

To exploit the statistical redundancy of quantisation indices we have proposed to employ the Huffman coding instead of the fixed-length coding used in DISCOVER. Currently nice gains in rate-distortion performance are observed. However, the problem of decoder synchronisation in a case of erroneous bit plane decoding, is not fully solved.

Conclusion and perspectives

Synthesis

In this thesis, we have addressed various problems related to distributed source coding, and to distributed video coding (DVC) in particular. The main objective of this study was the improvement of the rate-distortion performance of a real DVC system, DISCOVER, although other issues like architecture limitations, have also been discussed. The contributions are presented in three chapters, each corresponding to a particular aspect of DVC: side information extraction, correlation modeling, and coding of sources with memory. Here we make a synthesis of the study.

Side information extraction

In chapter 3 we have tried to give a vision on the side information (SI) extraction problem in Wyner-Ziv video coding. Different ways of improving the quality of the side information has been considered.

- A versatile mesh-based motion estimation has been considered for the SI extraction. In distributed video coding the complexity and the rate cost of the motion field don't matter, as the motion field need not be transmitted. Therefore, complex motion models can be used without any limitation if they improve the rate-distortion performance of the codec. However, our experiments have shown that the mesh-based motion estimation is not all the time superior to the block-based motion estimation in the DVC context, and only an adaptive combination of the two methods may give an advantage.
- We developed a "hybrid" approach for constructing a superior SI from multiple hypotheses available from several SI extraction methods, e.g. a mesh-based and a block-based methods. We have also considered the approach called "Multi-hypotheses decoding" in our study, and showed that our "hybrid" approach is comparable to this "multihypotheses" approach in terms of rate-distortion performance.

- The spatial side information has also been considered in this chapter. Our experiments show that the original WZ frame of lower resolution and/or coarsely quantised (encoded at low rate with an H.264-like coder in “intra” mode), is not better than the temporal side information probably containing interpolation errors, but having the full spatial resolution. A measure of the SI quality has been proposed, which is more correlated with the rate-distortion performance of the codec than the usually employed PSNR measure.
- Finally, we have treated the SI improvement problem as a denoising problem. We have tested two denoising methods, namely the wavelet thresholding, and the BM3D method. According to our preliminary tests the latter method allows for a slight improvement in the rate-distortion performance.

Correlation modeling in Wyner-Ziv coding

In chapter 4 we have addressed different problems concerning correlation modeling and model parameters estimation.

- The state-of-the-art decoder correlation model adopted in DISCOVER has been studied thoroughly, its drawbacks and weak points have been identified. An algorithm inspired by the EM (Expectation-Maximisation) approach for improving the correlation model estimation has been proposed.
- Closed-form expressions for the minimal mean-squared error reconstruction at the decoder have been derived for the Laplacian correlation model.
- We have presented a hybrid encoder/decoder rate control solution, which main goal is to reduce the decoding complexity by decreasing the number of decoder requests. The encoder part of this hybrid solution estimates the minimal rate, while the sufficient rate is determined by the decoder. The sufficient rate is determined using the developed bit error rate estimation technique, using turbo decoder log-likelihood ratios and a checksum verification for robust error detection.
- An algorithm for adaptive quantisation table design for Wyner-Ziv frames has been proposed. The quantisation table is optimised in terms of the rate-distortion performance with a constraint on the distortion imposed by adjacent key frames distortion. The proposed technique relies on the developed robust correlation model estimation approach based on partial motion estimation at the encoder.
- We have also considered a mixed intra/WZ coding mode for WZ frames, with a block-level decision inspired by the PRISM architecture. Unlike the approach in [TTT⁺06b], our method is designed in the transform domain.

Source coding aspects of Wyner-Ziv coding

In the last chapter 5 we treat the DVC as a Wyner-Ziv coding problem of sources with memory. We address the memory problem on four different levels.

- We used the oriented wavelet transform (OWT) by [CG06] instead of the DCT adopted in DISCOVER. We show that an appropriate transform (e.g. OWT) permits more efficient distributed coding of video data, just as in the case of the conventional video coding.
- We treat the memory inside a frequency band with the Distributed DPCM approach. We propose an algorithm for designing prediction filters for both the WZ data and the SI so that the correlation between prediction errors is preserved as much as possible.
- The memory on the level of quantisation indices is taken into account by a Huffman code (instead of the fixed-length code adopted in DISCOVER).

Perspectives

Although many problems brought up in this thesis, have been partly addressed by proposed algorithms, the research can be continued further to improve certain solutions, or to develop more efficient approaches based on the proposed ones. Below we list some general ideas that can be taken as starting points for future research directions.

- We have proposed a “hybrid” approach to combine two side information hypotheses (obtained using mesh-based and block-based motion estimation methods) for a more efficient decoding. In fact, this approach can be extended to the case of 3 and more available hypotheses, with inclusion of the spatial SI, denoised SI etc., resulting in an efficient complex method for SI extraction.
- We have not considered using the hash information for helping the temporal SI extraction. However, we think that transmitting a limited amount of supplementary information (hash) about the encoded WZ frame can be necessary for a good side information extraction method, because this information is required to synchronise the SI and the WZ frame, when the motion is not uniform in time. Moreover, with the “hybrid” approach, the hash information could play a double role in the SI extraction: first, it would help the *construction* of SI hypotheses by different methods, and second, would help the hybrid *combination* algorithm.
- The precise estimation of correlation model parameters is very important for an efficient DVC. Encoder and decoder estimations of the model could be combined

in a single estimation, which is refined during the decoding, and the refined value can be transmitted back to the encoder. Also, the refined value can be used by the hybrid SI extraction approach.

- In Section 5.1.2 we have shown that the DCT in the DISCOVER codec can be replaced by another transform, which improves its rate-distortion performance. The search for better transforms for DVC can therefore be continued.
- A more long-term perspective for DVC would be to define requirements to a practical DVC system, in certain target application fields (e.g. video surveillance). This will help concentrating research efforts in certain directions, removing certain (the most important for this field) limitations of the current DVC architecture, and eventually contribute to the spreading of DVC, and using it in practical systems.

Publications

International conferences:

- D. Kubasov, C. Guillemot.
“Mesh-based motion-compensated interpolation for side information extraction in distributed video coding,” *IEEE International Conference on Image Processing (ICIP)*, Atlanta, USA, Oct 2006.
- D. Kubasov, K. Lajnef, and C. Guillemot.
“A hybrid encoder/decoder rate control for Wyner-Ziv video coding with a feedback channel,” *IEEE 9th International Workshop on Multimedia Signal Processing (MMSP)*, pages 251-254, Chania, Crete, Greece, Oct 2007.
- D. Kubasov, J. Nayak, and C. Guillemot.
“Optimal reconstruction in Wyner-Ziv video coding with multiple side information¹,” *IEEE 9th International Workshop on Multimedia Signal Processing (MMSP)*, pages 183-186, Chania, Crete, Greece, Oct 2007.
- X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret.
“The discover codec: Architecture, techniques and evaluation,” *Picture Coding Symposium (PCS)*, Lisbon, Portugal, Nov 2007.
- J. Nayak, G. Rath, D. Kubasov, and C. Guillemot.
“Spatial prediction in distributed video coding using Wyner-Ziv DPCM,” *IEEE 10th International Workshop on Multimedia Signal Processing (MMSP)*, Cairns, Australia, Oct 2008.

¹Won “Best student paper award”

Glossaire/Glossary

<i>English</i>	<i>French (if used)</i>	
BCJR		Bahl, Cocke, Jelinek, and Raviv (<i>algorithm of</i>)
BER		Bit Error Rate
BSC	CBS	Binary Symmetric Channel
CABAC		Context Adaptative Binary Arithmetic Coder
CRC		Cyclic Redundancy Check
dB		Decibel
DCT		Discrete Cosine Transform
DPCM		Differential Pulse Code Modulation
DSC		Distributed Source Coding
DVC	CVD	Distributed Video Coding
FLC	CLF	Fixed Length Code
EM		Expectation-Maximisation
FSM		Finite State Machine
GOP		Group Of Pictures
HMM		Hidden Markov Model
IDCT		Inverse Discrete Cosine Transform
kb, kbit		kilobit(s)
kbps		kilobit(s) per second
LDPC		Low Density Parity Check <i>codes</i>
LLR		Log-Likelihood Ratio
MCTI		Motion Compensation Temporal Interpolation
ME		Motion Estimation
MSB		Most Significant Bit
MSE	EQM	Mean Square Error
OWT		Oriented Wavelet Transform
PSNR		Peak Signal to Noise Ratio
p.d.f.		probability density function
QCIF		Quarter Common Intermediate Format (<i>for video</i>)
QIndex		Quantisation Index (<i>refers to coding of WZ frames</i>)
QP		Quantisation Parameter (<i>refers to coding of key frames</i>)
RSC	CRS	Recursive Systematic Convolutional (<i>encoder</i>)
SI	IA	Side Information
SIQ		Side Information Quality <i>measure of</i>
SISO		Soft-Input-Soft-Output
SNR		Signal to noise ratio
SW		Slepian-Wolf
TCQ		Trellis Coded Quantisation
VLC	CLV	Variable Length Code
WZ		Wyner-Ziv

Bibliography

- [AABP08] J. Areia, J. Ascenso, C. Brites, and F. Pereira. Low complexity hybrid rate control for lower complexity Wyner-Ziv video decoding. In *Proc. of EURASIP European Signal Processing Conf. (EUSIPCO)*, Lausanne, Switzerland, Aug 2008.
- [AAD⁺07] Xavi Artigas, Joao Ascenso, Marco Dalai, Sven Klomp, Denis Kubasov, and Mourad Ouaret. The discover codec: Architecture, techniques and evaluation. *Picture Coding Symposium (PCS)*, Nov 2007.
- [ABB96] L. Alparone, M. Barni, and F. Bartolini. Adaptively weighted vector-median filters for motion-fields smoothing. In *Proc. Intl. Conf. Acc. Speech Signal Processing*, pages 2267–2270, 1996.
- [ABP05a] J. Ascenso, C. Brites, and F. Pereira. Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. In *Proc. of the 5th EURASIP Conf. on Speech, Image Processing, Multimedia Communications and Services*, Smolenice, Slovak Republic, 2005.
- [ABP05b] J. Ascenso, C. Brites, and F. Pereira. Motion compensated refinement for low complexity pixel based distributed video coding. In *Proc. of International Workshop on Very Low Bitrate Video Coding*, Sardinia, Italy, 2005.
- [ABP06] J. Ascenso, C. Brites, and F. Pereira. Content adaptive Wyner-Ziv video coding driven by motion activity. In *Proc. Intl. Conf. Image Processing*, pages 605 – 608, 2006.
- [ABP08] J. Ascenso, C. Brites, and F. Pereira. Design and performance of a novel low-density parity-check code for distributed video coding. In *Proc. Intl. Conf. Image Processing*, San Diego, California, USA, Oct 2008.
- [AFAW06] A. B. B. Adikari, W. A. C. Fernando, H. K. Arachchi, and W. A. R. Weerakkody. Wyner-Ziv coding with temporal and spatial correlations for motion video. In *Proceedings of Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1188 – 1191, 2006.

- [AG02] A. Aaron and B. Girod. Compression with side information using turbo codes. In *Proc. Data Compression Conf.*, pages 252–261, 2002.
- [AP07] J. Ascenso and F. Pereira. Adaptive hash-based side information exploitation for efficient Wyner-Ziv video coding. In *Proc. Intl. Conf. Image Processing*, volume 3, pages III–29–III–32, Oct 2007.
- [AP08] J. Ascenso and F. Pereira. Hierarchical motion estimation for side information creation in Wyner-Ziv video coding. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication (ICUIMC)*, pages 347–352, Suwon, Korea, 2008.
- [ARG04] A. Aaron, S. Rane, and B. Girod. Wyner-Ziv video coding with hash-based motion compensation at the receiver. In *Proc. Intl. Conf. Image Processing*, pages 24 – 27, 2004.
- [ARSG04] Anne Aaron, Shantanu Rane, Eric Setton, and Bernd Girod. Transform-domain Wyner-Ziv codec for video. In *Proc. SPIE Visual Communications and Image Processing*, pages 520–528, 2004.
- [ASG03] A. Aaron, E. Setton, and B. Girod. Towards practical Wyner-Ziv coding of video. In *Proc. Intl. Conf. Image Processing*, volume 3, pages III–869–72 vol.2, Sept 2003.
- [ATBS07] S. Argyropoulos, N. Thomos, N.V. Boulgouris, and M.G. Strintzis. Adaptive frame interpolation for Wyner-Ziv video coding. In *Proc. of the IEEE 9th Workshop on Multimedia Signal Processing*, pages 159–162, Chania, Crete, Greece, Oct 2007.
- [AZG02] A. Aaron, Rui Zhang, and B. Girod. Wyner-Ziv coding of motion video. In *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 240–244, Nov 2002.
- [BAP06a] C. Brites, J. Ascenso, and F. Pereira. Improving transform domain Wyner-Ziv video coding performance. *Proc. Intl. Conf. Acc. Speech Signal Processing*, 2, May 2006.
- [BAP06b] C. Brites, J. Ascenso, and F. Pereira. Modeling correlation noise statistics at decoder for pixel based Wyner-Ziv video coding. In *Proc. of Picture Coding Symposium (PCS)*, Beijing, China, Apr 2006.
- [BAP06c] C. Brites, J. Ascenso, and F. Pereira. Studying temporal correlation noise modeling for pixel based Wyner-Ziv video coding. *Proc. Intl. Conf. Image Processing*, pages 273–276, Oct. 2006.

- [BCJR74] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, 20(2):284–287, Mar 1974.
- [Ber72] T. Berger. *Rate-distortion theory: A mathematical basis for data compression*. Prentice-Hall, 1972.
- [BG96] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Trans. Commun.*, 44(10):1261–1271, Oct 1996.
- [BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: turbo-codes. In *Conference Record of IEEE International Conference on Communications*, volume 2, pages 1064–1070, Geneva, May 1993.
- [BM01] J. Bajcsy and P. Mitran. Coding for the Slepian-Wolf problem with turbo codes. In *Proc. of IEEE Global Telecommunications Conference (GLOBE-COM)*, volume 2, pages 1400–1404, 2001.
- [BP07] C. Brites and F. Pereira. Encoder rate control for transform domain Wyner-Ziv video coding. In *Proc. Intl. Conf. Image Processing*, volume 2, pages 5–8, Oct 2007.
- [BP08] C. Brites and F. Pereira. Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding. *IEEE Trans. Circuits Syst. Video Technol.*, 2008. accepted.
- [CEGK98] G. Cote, B. Erol, M. Gallant, and F. Kossentini. H.263+: video coding at low bit rates. *IEEE Trans. Circuits Syst. Video Technol.*, 8(7):849–866, Nov 1998.
- [CG06] V. Chappelier and C. Guillemot. Oriented wavelet transform for image compression and denoising. *IEEE Trans. Image Processing*, 15(10):2892–2903, Oct. 2006.
- [CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, 1991.
- [CYV00] S.G. Chang, Bin Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Processing*, 9(9):1532–1546, Sep 2000.
- [Dau92] Ingrid Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

- [dBvKOS00] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Janvier 2000.
- [DFKE07] K.. Dabov, A.. Foi, V.. Katkovnik, and K.. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Processing*, 16(8):2080–2095, Aug 2007.
- [dis] DISCOVER project web page. <http://www.discoverdvc.org/>.
- [dis07] DISCOVER codec performance evaluation web-page, 2007. <http://www.img.lx.it.pt/~discover/home.html>.
- [DS98] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.
- [FE01] Michael Fleming and Michelle Effros. Network vector quantization. In *Proc. Data Compression Conf.*, page 13, Washington, DC, USA, 2001. IEEE Computer Society.
- [FKE07] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Trans. Image Processing*, 16(5):1395–1411, May 2007.
- [Foi] A. Foi. Block-matching and 3D filtering (BM3D) algorithm web page: software, publications, experimental data, presentations, test-images, etc. <http://www.cs.tut.fi/foi/GCF-BM3D/>.
- [FZE04] M. Fleming, Qian Zhao, and M. Effros. Network vector quantization. *IEEE Trans. Inform. Theory*, 50(8):1584–1604, Aug. 2004.
- [Gal62] R. G. Gallager. Low density parity check codes. *Trans. IRE Prof. Group on Inform. Theory*, 8:21–28, Jan. 1962.
- [Gal02] Franck Galpin. *Représentation 3D de séquences vidéo. Schéma d'extraction automatique d'un flux de modèles 3D, application à la compression et à la réalité virtuelle*. PhD thesis, Université de Rennes 1, 2002.
- [GARRM05] B. Girod, A.M. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, Jan 2005.
- [GFZ01] J. Garcia-Frias and Ying Zhao. Compression of correlated binary sources using turbo codes. *IEEE Communications Letters*, 5(10):417–419, Oct 2001.
- [GN98] R.M. Gray and D.L. Neuhoff. Quantization. *IEEE Trans. Inform. Theory*, 44(6):2325–2383, Oct 1998.

- [Huf52] D. Huffman. A method for the construction of minimum redundancy codes. In *Proc. of the IRE*, volume 40, pages 1098–1101, 1952.
- [JAI97] A. Kh. Al Jabri and S. Al-Issa. Zero-error codes for correlated information sources. In *Proc. of IMA Int. Conf.*, volume 1355, pages 17–22, 1997.
- [KG06] Denis Kubasov and Christine Guillemot. Mesh-based motion-compensated interpolation for side information extraction in distributed video coding. In *Proc. Intl. Conf. Image Processing*, Atlanta, USA, Oct 2006.
- [KLG07] D. Kubasov, K. Lajnef, and C. Guillemot. A hybrid encoder/decoder rate control for Wyner-Ziv video coding with a feedback channel. In *Proc. of IEEE 9th Workshop on Multimedia Signal Processing (MMSP)*, pages 251–254, Oct 2007.
- [Kli04] Dan Klien. Lagrange multipliers without permanent scarring. <http://www.cs.berkeley.edu/~klein/papers/lagrange-multipliers.pdf>, August 2004.
- [KNG07] D. Kubasov, J. Nayak, and C. Guillemot. Optimal reconstruction in Wyner-Ziv video coding with multiple side information. In *Proc. of IEEE 9th Workshop on Multimedia Signal Processing (MMSP)*, pages 183–186, Oct 2007.
- [KVO06] Sven Klomp, Yuri Vatis, and Joern Ostermann. Side information interpolation with sub-pel motion compensation for Wyner-Ziv decoder. In *Proc. Int. Conf. on Signal Processing and Multimedia Applications*, volume 0, pages 178–182, aug 2006.
- [Laj06] Khaled Lajnef. *Étude du codage de sources distribuées pour de nouveaux concepts en compression vidéo*. PhD thesis, Université de Rennes 1, 2006.
- [Lau00] N. Laurent. Hierarchical mesh-based global motion estimation, including occlusion areas detection. In *Proc. Intl. Conf. Image Processing*, volume 3, pages 620–623, 2000.
- [LCLX04] Zhixin Liu, Samuel Cheng, Angelos D. Liveris, and Zixiang Xiong. Slepian-Wolf coded nested quantization (SWC-NQ) for Wyner-Ziv coding: performance analysis and code design. In *Proc. Data Compression Conf.*, page 322, Washington, DC, USA, 2004. IEEE Computer Society.
- [LD05] Zhen Li and E.J. Delp. Wyner-Ziv video side estimator: conventional motion search methods revisited. In *Proc. Intl. Conf. Image Processing*, volume 1, pages I-825–8, Sept 2005.

- [LLD07] Z. Li, L. Liu, and E. J. Delp. Rate distortion analysis of motion side estimation in Wyner-Ziv video coding. *IEEE Trans. Image Processing*, 16(1):98–113, Jan 2007.
- [Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28(2):129–137, Mar 1982.
- [LXG02] A.D. Liveris, Zixiang Xiong, and C.N. Georgiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, 6(10):440–442, Oct 2002.
- [Mac99] D MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inform. Theory*, (45):399–431, 1999.
- [Mac05] D.J.C. MacKay. Fountain codes. *IEE Proceedings - Communications*, 152(6):1062–1068, Dec 2005.
- [Mar00] Gwenaëlle Marquant. *Représentations par maillage adaptif déformable pour la manipulation et la communication d’objets vidéo*. PhD thesis, Université de Rennes 1, 2000.
- [MdQM07] B. Macchiavello, R.L. de Queiroz, and D. Mukherjee. Motion-based side-information generation for a scalable Wyner-Ziv video coder. In *Proc. Intl. Conf. Image Processing*, volume 6, pages VI–413–VI–416, Oct 2007.
- [MF90] M.W. Marcellin and T.R. Fischer. Trellis coded quantization of memoryless and Gauss-Markov sources. *IEEE Trans. Commun.*, 38(1):82–93, Jan 1990.
- [MGM07] M. Maitre, C. Guillemot, and L. Morin. 3-D model-based frame interpolation for distributed video coding of static scenes. *IEEE Trans. Image Processing*, 16(5):1246–1257, May 2007.
- [MHKK03] H.S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky. Low-complexity transform and quantization in H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):598–603, July 2003.
- [MKR05] K. Misra, S. Karande, and H. Radha. Multi-hypothesis distributed video coding using LDPC codes. In *Proc. of 43rd Annual Allerton Conference on Communication, Control and Computing*, Monticello, USA, 2005.
- [Moo96] T. K. Moon. The expectation-maximization algorithm. *IEEE Trans. Signal Processing*, 13(6):47–60, 1996.
- [MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North Holland, 1977.

- [MSW03] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):620 – 636, 2003.
- [Nol78] P. Noll. On predictive quantizing schemes. *Bell System Technical Journal*, (57):1499–1532, 1978.
- [NRG08] J. Nayak, G. Rath, and C. Guillemot. Prediction filter desing for Wyner-Ziv coding of sources with memory. *IEEE Trans. Inform. Theory*, Feb 2008. submitted.
- [NRKG08] J. Nayak, G. Rath, D. Kubasov, and C. Guillemot. Spatial prediction in distributed video coding using Wyner-Ziv DPCM. In *Proc. of IEEE 10th Workshop on Multimedia Signal Processing (MMSP)*, Cairns, Queensland, Australia, Oct 2008.
- [OMK⁺06] N. Ohkubo, N. Miki, Y. Kishiyama, K. Higuchi, and M. Sawahashi. Performance comparison between turbo code and rate-compatible LDPC code for evolved ultra downlink OFDM radio access. In *Military Communications Conference (MILCOM)*, pages 1–7, Oct 2006.
- [Pas76] R. C. Pasco. *Source Coding Algorithms for Fast Data Compression*. PhD thesis, Stanford University, CA, may 1976.
- [PCR03] S.S. Pradhan, J. Chou, and K. Ramchandran. Duality between source coding and channel coding and its extension to the side information case. *IEEE Trans. Inform. Theory*, 49(5):1181–1203, May 2003.
- [PM92] William B. Pennebaker and Joan L. Mitchell. *JPEG Still Image Data Compression Standard*. Kluwer Academic Publishers, Norwell, MA, USA, 1992.
- [PMR07] R. Puri, A. Majumdar, and K. Ramchandran. PRISM: A video coding paradigm with motion estimation at the decoder. *IEEE Trans. Image Processing*, 16(10):2436–2448, Oct 2007.
- [PR99] S.S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): design and construction. In *Proc. Data Compression Conf.*, pages 158–167, Mar 1999.
- [PR02] R. Puri and K. Ramchandran. PRISM: A new robust video coding architecture based on distributed compression principles. In *Proc. of the 40th Allerton Conference on Communication, Control and Computing*, Allerton, IL, USA, Oct 2002.

- [PR03a] S.S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): design and construction. *IEEE Trans. Inform. Theory*, 49(3):626–643, Mar 2003.
- [PR03b] R. Puri and Kannan Ramchandran. PRISM: A video coding architecture based on distributed compression principles. Technical Report UCB/ERL M03/6, EECS Department, University of California, Berkeley, 2003.
- [Ris76] J.J. Rissanen. Generalized Kraft inequality and arithmetic coding. *J. Research and Development*, 20(3):198–203, 1976.
- [RMZG03] David Rebollo-Monedero, Rui Zhang, and Bernd Girod. Design of optimal quantizers for distributed source coding. In *Proc. Data Compression Conf.*, page 13, Washington, DC, USA, 2003. IEEE Computer Society.
- [RY90] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithm, Advantages, and Applications*. New York: Academic, 1990.
- [SCE01] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The jpeg 2000 still image compression standard. *IEEE Signal processing Magazine*, 18:36–58, 2001.
- [Ser00] S.D. Servetto. Lattice quantization with side information. In *Proc. Data Compression Conf.*, pages 510–519, 2000.
- [Ser07] S.D. Servetto. Lattice quantization with side information: codes, asymptotics, and applications in sensor networks. *IEEE Trans. Inform. Theory*, 53(2):714–731, Feb 2007.
- [Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423 623–656, 1948.
- [SW73] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, 19(4):471–480, 1973.
- [SXJLW07] Fang Sheng, Li Xu-Jian, and Zhang Li-Wei. A Lloyd-Max-based non-uniform quantization scheme for distributed video coding. *Proc. of 8th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 1:848–853, Aug 2007.
- [TGFZ03] Tao Tian, J. Garcia-Frias, and Wei Zhong. Compression of correlated sources using LDPC codes. In *Proc. Data Compression Conf.*, pages 450–, March 2003.
- [Tja00] T. Tjalkens. The complexity of minimum redundancy coding. In *Proc. Intl. Conf. Inform. Theory*, page 373, June 2000.

- [TPPT07] M. Tagliasacchi, J. Pedro, F. Pereira, and S. Tubaro. An efficient request stopping method at the turbo decoder in distributed video coding. In *Proc. of EURASIP European Signal Processing Conf. (EUSIPCO)*, pages 1–4, Poznan, Poland, Sep 2007.
- [TTES96] C. Toklu, A.M. Tekalp, A.T. Erdem, and M.I. Sezan. 2-D mesh-based tracking of deformable objects with occlusion. In *Proc. Intl. Conf. Image Processing*, volume 1, pages 933–936, Sep 1996.
- [TTT⁺06a] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira. Exploiting spatial redundancy in pixel domain Wyner-Ziv video coding. In *Proc. Intl. Conf. Image Processing*, pages 253 – 256, 2006.
- [TTT⁺06b] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, and F. Pereira. Intra mode decision based on spatio-temporal cues in pixel domain Wyner-Ziv video coding. In *Proc. Intl. Conf. Acc. Speech Signal Processing*, volume 2, May 2006.
- [VAG05] D. Varodayan, A. Aaron, and B. Girod. Rate-adaptive distributed source coding using low-density parity-check codes. In *Conference Record of the 39th Asilomar Conference on Signals, Systems and Computers*, pages 1203–1207, Nov 2005.
- [VAG06] David Varodayan, Anne Aaron, and Bernd Girod. Rate-adaptive codes for distributed source coding. *EURASIP Signal Process. J., special issue on distributed source coding*, 86(11):3123–3130, 2006.
- [VCFG08] D. Varodayan, D. Chen, M. Flierl, and B. Girod. Wyner-Ziv coding of video with unsupervised motion vector learning. *Signal Processing: Image Communication*, 23(5):369 – 378, 2008.
- [VKO07] Y. Vatis, S. Klomp, and J. Ostermann. Enhanced reconstruction of the quantized transform coefficients for Wyner-Ziv coding. In *Proceedings of Int. Conf. on Multimedia & Expo (ICME)*, Beijing, China, July 2007.
- [WBSS04] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, Apr 2004.
- [Wil] Ross N. Williams. A painless guide to CRC error detection algorithms. http://www.repairfaq.org/filipg/LINK/F_crc_v3.html.
- [WNC87] I Witten, R Neal, and J Cleary. Arithmetic coding for data compression. *Communications of the ACM*, (30), 1987.

- [WSBL03] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):560–576, 2003.
- [Wyn74] A. D. Wyner. Recent results in the shannon theory. *IEEE Trans. Inform. Theory*, (20), 1974.
- [WZ76] A. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inform. Theory*, 22(1):1–10, 1976.
- [XLCL03] Zixiang Xiong, A.D. Liveris, S. Cheng, and Zhixin Liu. Nested quantization and Slepian-Wolf coding: a Wyner-Ziv coding paradigm for i.i.d. sources. In *Proc. of IEEE Workshop on Statistical Signal Processing*, pages 399–402, Sept.-1 Oct. 2003.
- [XROZ99] Zixiang Xiong, K. Ramchandran, M.T. Orchard, and Ya-Qin Zhang. A comparative study of DCT- and wavelet-based image coding. *IEEE Trans. Circuits Syst. Video Technol.*, 9(5):692–695, Aug 1999.
- [XSX07] Qian Xu, V. Stankovic, and Zixiang Xiong. Distributed joint source-channel coding of video using Raptor codes. *IEEE J. Select. Areas Commun.*, 25(4):851–861, May 2007.
- [YB00] Ying-On Yan and T. Berger. On instantaneous codes for zero-error coding of two correlated sources. In *Proc. Intl. Conf. Inform. Theory*, pages 344–, 2000.
- [ZE01] Qian Zhao and M. Effros. Optimal code design for lossless and near lossless source coding in multiple access networks. In *Proceedings of Data Compression Conference (DCC)*, pages 263–272, 2001.
- [ZE03] Qian Zhao and M. Effros. Lossless and near-lossless source coding for multiple access networks. *IEEE Trans. Inform. Theory*, 49(1):112–128, Jan 2003.
- [ZS98] R. Zamir and S. Shamai. Nested linear/lattice codes for Wyner-Ziv encoding. In *Proc. of Information Theory Workshop*, pages 92–93, Jun 1998.
- [ZSE02] R. Zamir, S. Shamai, and U. Erez. Nested linear/lattice codes for structured multiterminal binning. *IEEE Trans. Inform. Theory*, 48(6):1250–1276, Jun 2002.

List of Figures

1.1	Relation entre l'entropie et l'information mutuelle	16
1.2	Schéma générale de compression avec pertes	20
1.3	La fonction débit-distorsion pour une source Gaussienne $\mathcal{N}(0, \sigma^2)$	23
1.4	Schéma générale d'un système de communication	26
1.5	Modèles des canaux	27
1.6	Schéma d'un codeur turbo	27
1.7	Exemple d'un système de codage de sources distribuées	29
1.8	Illustration pour le théorème de Slepian et Wolf	30
1.9	Région réalisable de débit pour le codage de Slepian-Wolf	31
1.10	Illustration pour le théorème de Wyner et Ziv	31
1.11	Illustration du rôle des variables X, Y, Z et \hat{X}	32
2.1	Codage de Slepian-Wolf avec information adjacente	36
2.2	Une partition $\mathcal{P}(\mathcal{X})$ et le CLV associé	40
2.3	Illustration pour l'exemple de codage avec DISCUS	42
2.4	Schéma de codage de Slepian-Wolf par Garcia-Frias et Zhao	44
2.5	Exemple d'une machine à nombre d'états fini (4 états)	45
2.6	Schéma de codage de Slepian-Wolf par Aaron et Girod	46
2.7	Graphe "bipartite" du code LDPC	47
2.8	Algorithme de passage des messages	48
2.9	Schéma d'un codeur de Wyner-Ziv	49
2.10	Illustration pour l'exemple du codage de Wyner-Ziv	50
2.11	Exemple d'une paire de lattices emboîtées	51
2.12	Schéma de codage de Zamir <i>et al.</i>	52
2.13	Schéma de codage de Wyner-Ziv DISCUS	53
2.14	Schéma de codage de Wyner-Ziv par Aaron et Girod	55
2.15	Architecture de PRISM	59
2.16	Représentation des plans de bits pour un vecteur de 64 coefficients. Les plans de bits sont arrangés dans l'ordre croissant de l'importance : le plan de bits 0 groupe les bits les moins importants, le plan de bits <i>max</i> - les bits les plus importants des coefficients c_0, c_1, \dots, c_{63}	61

2.17	Architecture de Stanford (domaine transformé)	63
2.18	Le schéma du système de codage vidéo DISCOVER	71
2.19	Le schéma de l'extraction de l'information adjacente dans DISCOVER	77
2.20	Obtention de vecteurs bidirectionnels à partir des vecteurs initiaux	78
3.1	Side information construction scheme (GOP of size 2)	90
3.2	Detailed scheme of the WZ decoder	92
3.3	2D regular triangular mesh	93
3.4	Hierarchical multi-resolution approach to minimisation	95
3.5	Rate-distortion performance of the WZ codec; key frames are compressed without losses	98
3.6	Rate-distortion performance of the WZ codec; key frames compressed to have approximately the same quality as WZ frames	99
3.7	Side information quality as function of key frames quantisation parameter QP	99
3.8	Problem with occlusions	100
3.9	Scheme of the WZ decoder with hybrid SI extraction in pixel domain	101
3.10	Example of hybrid SI extraction for Foreman sequence	103
3.11	Scheme of the WZ decoder with hybrid SI extraction in transform domain	104
3.12	Scheme of the WZ decoder with hybrid SI extraction in transform domain updated during decoding	106
3.13	Simple reconstruction with uniform coefficient distribution	107
3.14	Simple intermediate reconstruction with uniform coefficient distribution	108
3.15	Hybrid SI extraction using oracle	110
3.16	Rate-distortion performance of the WZ codec with Hybrid SI extraction	111
3.17	Multi-hypotheses WZ decoder scheme	113
3.18	Rate-distortion performance of the WZ codec with Multihypotheses decoding in the transform domain	116
3.19	Evolution of the temporal SI in time for different sequences and compression options	120
3.20	Temporal and spatial SI for the frame #2 of Soccer sequence	121
3.21	Correlation model analysis (DC and AC bands)	122
3.22	Log-likelihood ratio as function of $\Pr(x = 0)$	124
3.23	SIQ for temporal and spatial SI for the frame #2 of Soccer sequence	125
3.24	Frames selected for denoising experiments	128
3.25	Frames with synthetic Laplacian noise added	129
3.26	Denoised frames (synthetic Laplacian noise)	129
3.27	Example frames of side information	131
3.28	Rate-distortion performance with BM3D denoising method	134
4.1	Relationship between different correlation models	140
4.2	Example of correlation models for the frame #2 of Foreman, QCIF	141

4.3	Rate-distortion performance with different correlation models	142
4.4	Turbo decoding with on-line parameter learning	145
4.5	Correlation model parameter learning during turbo decoding	146
4.6	Rate-distortion performance with correlation model learning	147
4.7	Optimal reconstruction functions for different α	151
4.8	Rate-distortion performance with optimal MMSE reconstruction	153
4.9	Possible ways of obtaining α at the encoder side	156
4.10	Estimated minimal rate vs. real decoder rate per bit plane	158
4.11	Average number of decoder requests per frame	159
4.12	Artifacts due to the wrong estimation of the bit plane residual error rate	160
4.13	Rate-distortion performance with confidence measure and CRC	162
4.14	Rate-distortion performance with adaptive quantisation table for WZ frames	168
4.15	Frame by frame PSNR of the decoded sequence	169
4.16	Global correlation model parameter, "true" vs. decoder estimation . . .	170
4.17	Model parameter estimation using partial SI extraction	171
4.18	Correlation model parameter estimation error	172
4.19	DISCOVER codec with mixed "intra"/WZ encoding mode decision . . .	173
4.20	An example of frame partitioning	174
4.21	Experiments for choosing T_{rat} threshold	176
4.22	Rate-distortion performance with mixed "intra"/WZ encoding mode . .	180
4.23	Decoding time with mixed "intra"/WZ encoding mode	180
5.1	Quincunx decomposition of the grid L (two levels shown)	185
5.2	Rate-distortion performance with oriented wavelet transform (OWT) and DCT (optimised quantisation table for WZ frames is used)	188
5.3	DPCM coding scheme	190
5.4	WZDPCM coding scheme	191
5.5	DISCOVER codec with distributed DPCM on the DC band	193
5.6	Rate-distortion performance with WZDPCM for a Gauss-Markov process	195
5.7	Rate-distortion performance with WZDPCM for Foreman sequence . . .	196
5.8	Bit plane extraction, FLC	197
5.9	Bit plane compaction	197
5.10	Rate-distortion performance with FLC and Huffman coding of quanti- sation indices	199

Abstract

In this thesis we study different aspects of the distributed video coding (DVC), a new paradigm allowing flexible encoder/decoder complexity balance. Practical solutions offering better rate-distortion performances than existing algorithms, are proposed.

Several new approaches to the side information (SI) extraction in DVC are presented. We start by proposing a mesh-based SI extraction approach for more efficient motion estimation. After that, we develop a hybrid technique which allows combining multiple SI to obtain a SI of a better quality, and compare it to the “multihypotheses” decoding approach. We also study the spatial SI usage problem, and this lead us to deriving a new measure of the SI quality, which is more relevant to the decoding performance than the PSNR measure. Regarding the SI improvement problem as a denoising problem, we use this measure for evaluating performance gains brought by different denoising methods applied to the SI.

Problems of the correlation modeling between the SI and the WZ frame are addressed in the next chapter. We propose two methods for improving the correlation model parameters estimation. The first one is used for the model parameter refinement with the Expectation-Maximisation technique during the iterative decoding, while the second one can be employed when a precise correlation estimation is needed at the encoder. After that, we propose a hybrid encoder/decoder rate control solution, reducing significantly the decoder complexity and providing a robust decoder bit error rate estimation technique.

We also consider the rate-control at the encoder side. First, we design quantisation tables for WZ frames taking into account the actual level of the correlation in the temporal domain. Second, we encode blocks having low temporal correlation, with the key frame codec in “intra” mode.

In the last chapter, we study source coding aspects of DVC. We show that the oriented wavelet transform allows for a better energy compaction in WZ frames than the DCT, and thus for a more efficient coding of WZ frames. After that, we treat the memory on the frequency bands level by extending the classical DPCM approach to the case of the distributed coding (Distributed DPCM). Finally, the statistical redundancy in quantisation indices is removed by a Huffman code.

Résumé

Dans cette thèse, nous considérons différents problèmes du codage vidéo distribué, le nouveau paradigme de codage qui permet une allocation flexible de complexité entre le codeur et le décodeur. Nous proposons plusieurs outils afin d'améliorer les performances débit-distorsion des systèmes réels de codage.

Nous commençons par une étude sur l'extraction de l'information adjacente (IA). Nous considérons dans un premier temps une méthode d'estimation de mouvement basée sur une représentation par maillages déformables, pour l'extraction de l'IA. Ensuite, nous développons une approche hybride qui permet de combiner plusieurs IA pour un décodage plus performant, et la comparons avec la méthode dite de "décodage multihypothèse". Nous abordons également le problème d'utilisation de l'IA spatiale, ce qui nous permet de déduire une mesure de qualité pour l'IA. Nous utilisons ensuite cette mesure pour évaluer l'apport des méthodes de débruitage d'images appliquées à l'IA temporelle.

Les problèmes liés au modèle de corrélation entre l'IA et l'image WZ, sont abordés dans le chapitre suivant. Dans un premier temps, deux méthodes sont proposées pour améliorer l'estimation des paramètres du modèle. La première sert à raffiner l'estimation des paramètres pendant le décodage itératif à l'aide de la technique Expectation-Maximisation. La deuxième est utilisée par le codeur pour obtenir une estimation plus précise que celle que l'on peut obtenir au décodeur. Ensuite, une approche hybride de contrôle de débit est proposée, où l'estimation du débit est faite à la fois au codeur et au décodeur. Cette solution permet d'effectuer un contrôle assez fin, tout en diminuant le nombre de requêtes de bits supplémentaires et ainsi la complexité du décodeur.

Nous considérons également l'exploitation de la corrélation au codeur. Premièrement, nous optimisons le nombre de niveaux de quantification des images WZ en fonction du niveau de corrélation temporelle. Deuxièmement, nous codons certains blocs d'une image WZ avec un codeur conventionnel, si le niveau estimé de corrélation temporelle ne permet pas de garantir un codage efficace en mode WZ.

Les aspects du codage de source sont abordés dans le dernier chapitre. Dans un premier temps, nous montrons que la transformée par ondelettes orientées permet de séparer les fréquences dans une image WZ d'une manière plus efficace que la transformée DCT, et ainsi de la coder plus efficacement. Ensuite, nous étendons l'algorithme DPCM au cas du codage distribué, pour exploiter la mémoire restant dans la bande de basse fréquences de l'image transformée, en faisant une prédiction des coefficients simultanément dans la bande de l'image WZ et de l'IA. Enfin, pour exploiter la redondance statistique dans les indices de quantification, il est proposé de les coder avec un code de Huffman au lieu d'un code à longueur fixe.